Modelowanie i ranking

- Wykłady opracowano w oparciu o książkę Ricardo Baeza-Yates, Berthier Ribeiro-Neto, „*Modern Information Retrieval, the concepts and technology behind search*” 2nd edition, ACM Press Books, 2011
- Z tego samego źródła zaczerpnięto także różne zadania i przykłady wykorzystywane w treści wykładu.

Modelowanie i ranking

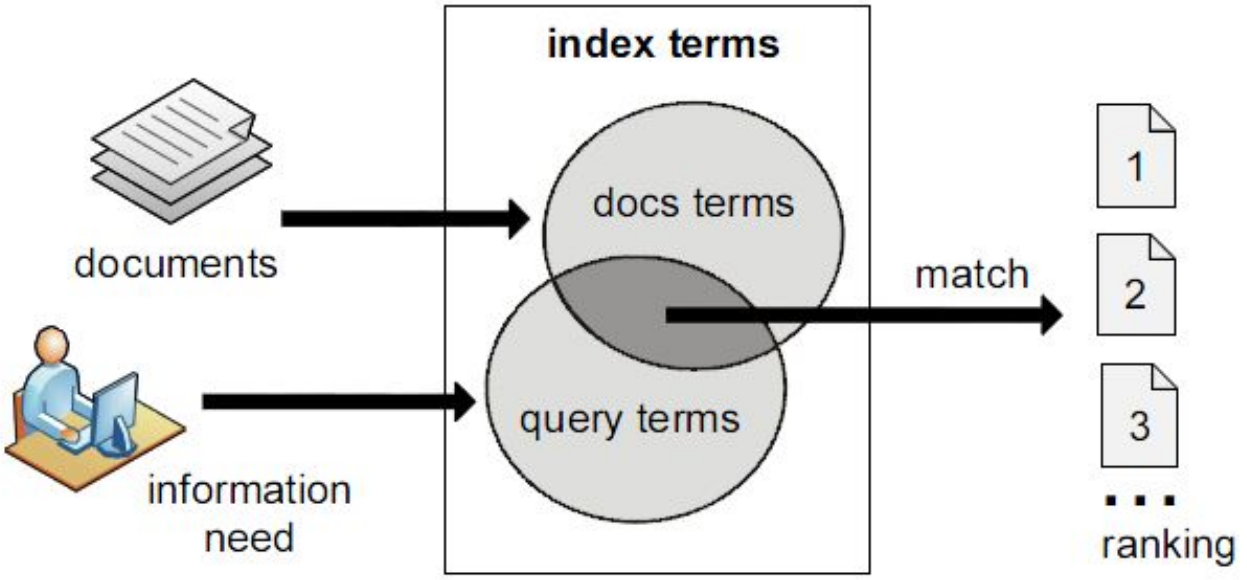
- Modelowanie w wyszukiwaniu informacji ma na celu określenie sposobu budowania funkcji rankingu
  - **Funkcja rankingu:** przypisuje określoną ocenę ilościową do dokumentu w odniesieniu do danego zapytania
- Modelowanie obejmuje dwa główne zadania:
  - Ustalenie założeń logicznych dla reprezentacji dokumentów i zapytań
  - Definicję funkcji rankingu pozwalającej na ilościowe wyrażenie podobieństwa zapytań i dokumentów zwracanych w odpowiedzi
- Do indeksowania i wyszukiwania dokumentów służą termy indeksujące

Model IR

- Term indeksujący:
  - W sensie ścisłym – jest to słowo kluczowe o pewnym znaczeniu, zazwyczaj rzeczownik
  - W sensie ogólnym – dowolne słowo pojawiające się w dokumencie
- Wyszukiwanie danych opiera się na termach indeksujących

Modelowanie i ranking

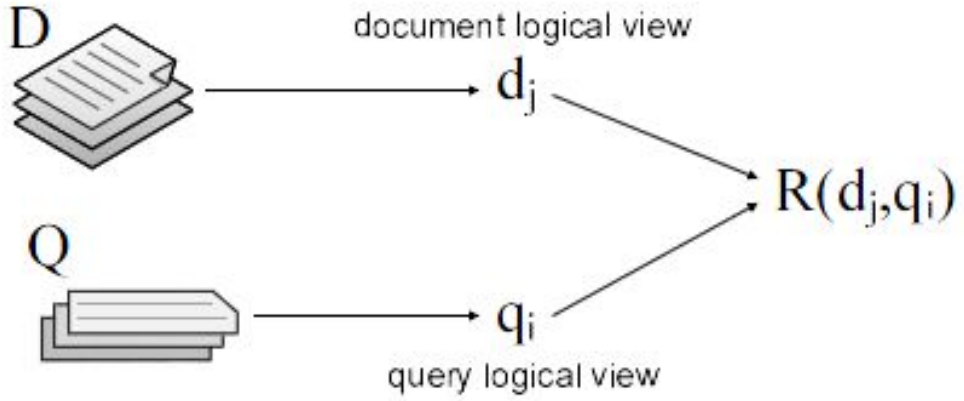
- Proces wyszukiwania informacji



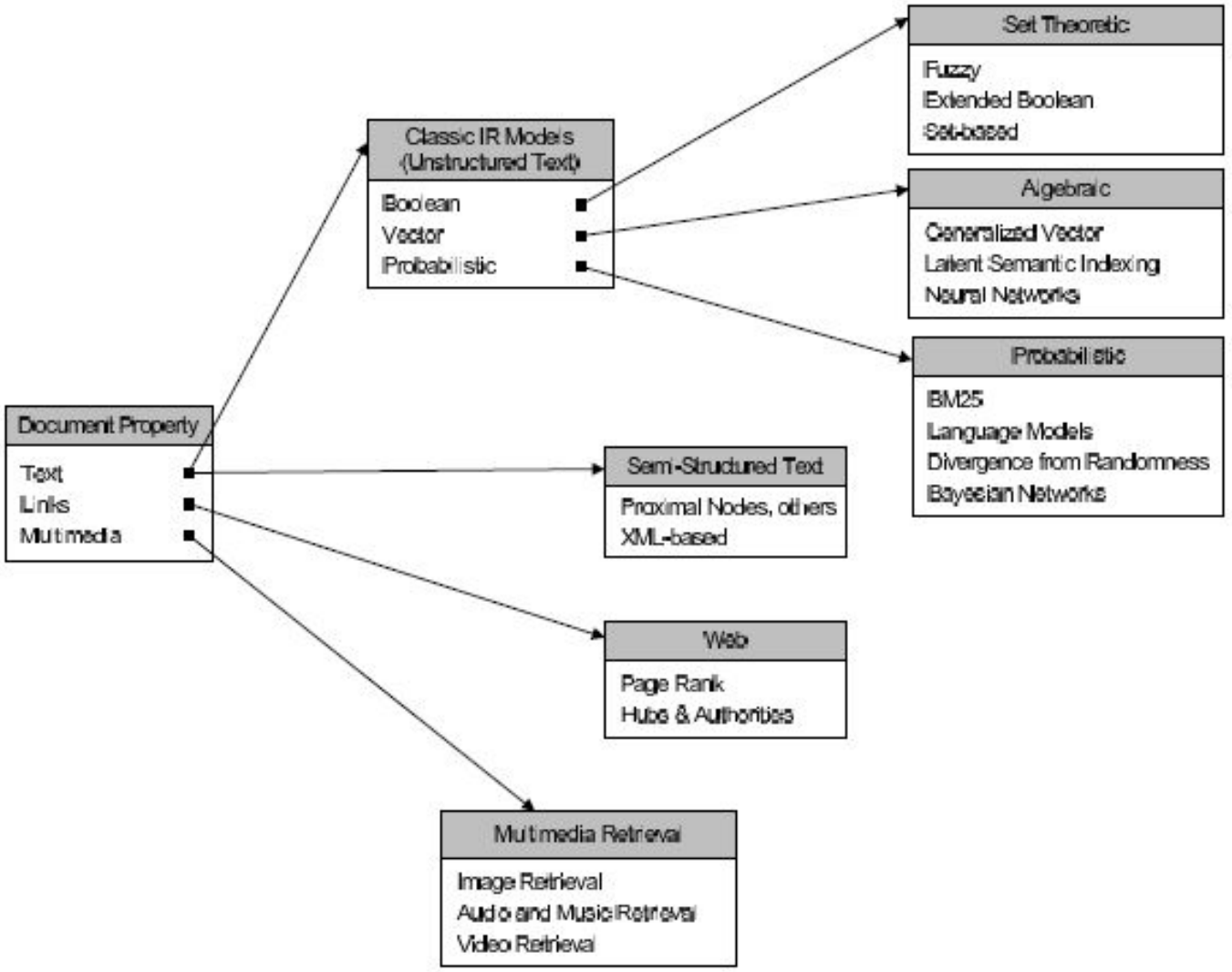
- **Ranking** to uporządkowanie dokumentów odzwierciedlające ich odpowiedniość względem zapytania
- System IR przewiduje które dokumenty są istotne dla użytkownika, z pewnym stopniem niepewności

Model IR

- Model IR jest czwórką $[D, Q, F, R(q_i, d_j)]$ gdzie:
  - *D* – zbiór logicznych perspektyw dokumentów
  - *Q* – zbiór logicznych perspektyw zapytań
  - *F* – założenia modelowania dokumentów i zapytań

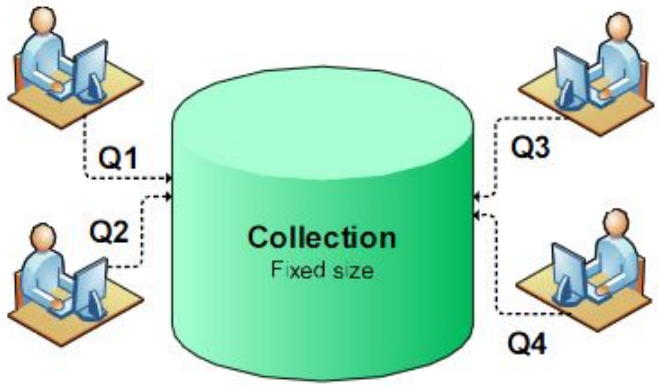


Klasyfikacja modeli

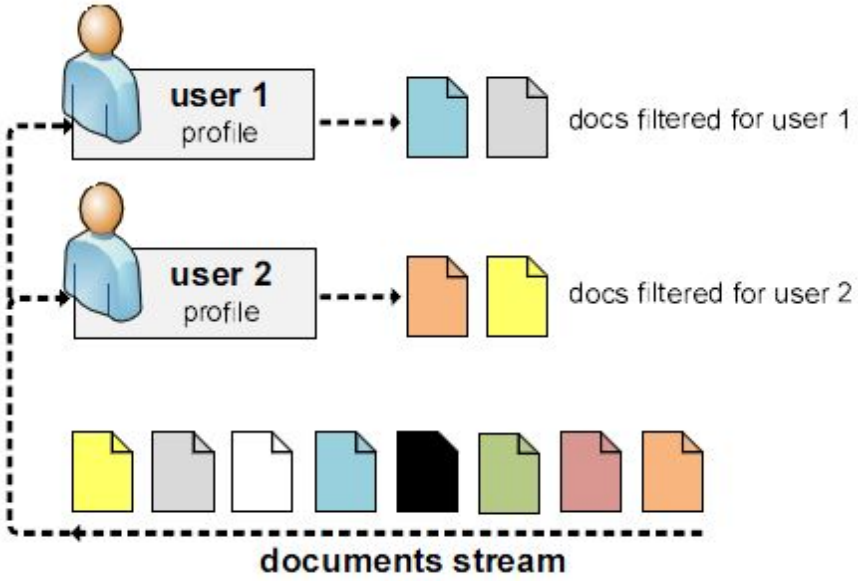


Wyszukiwanie ad-hoc i filtracja

- Każdy dokument jest reprezentowany przez termy indeksujące – słowa lub ciągi słów



- Wybrany zbiór termów reprezentuje dokument



Pojęcia podstawowe

- Słownik $V=\{k_1,...,k_t\}$ - zbiór różnych termów indeksujących
  - t - liczba termów w kolekcji dokumentów
  - k_i – elementarny term indeksujący

- $V=$ | k_1 | k_2 | k_3 | ... | k_t |

1	0	0	...	0
⋮				
1	1	1	...	1

pattern that represents documents (and queries) with the term k_1 and no other

pattern that represents documents (and queries) with all index terms

- Każdy z powyższych wzorców współwystępowania termów jest **koniunktywnym komponentem termów**

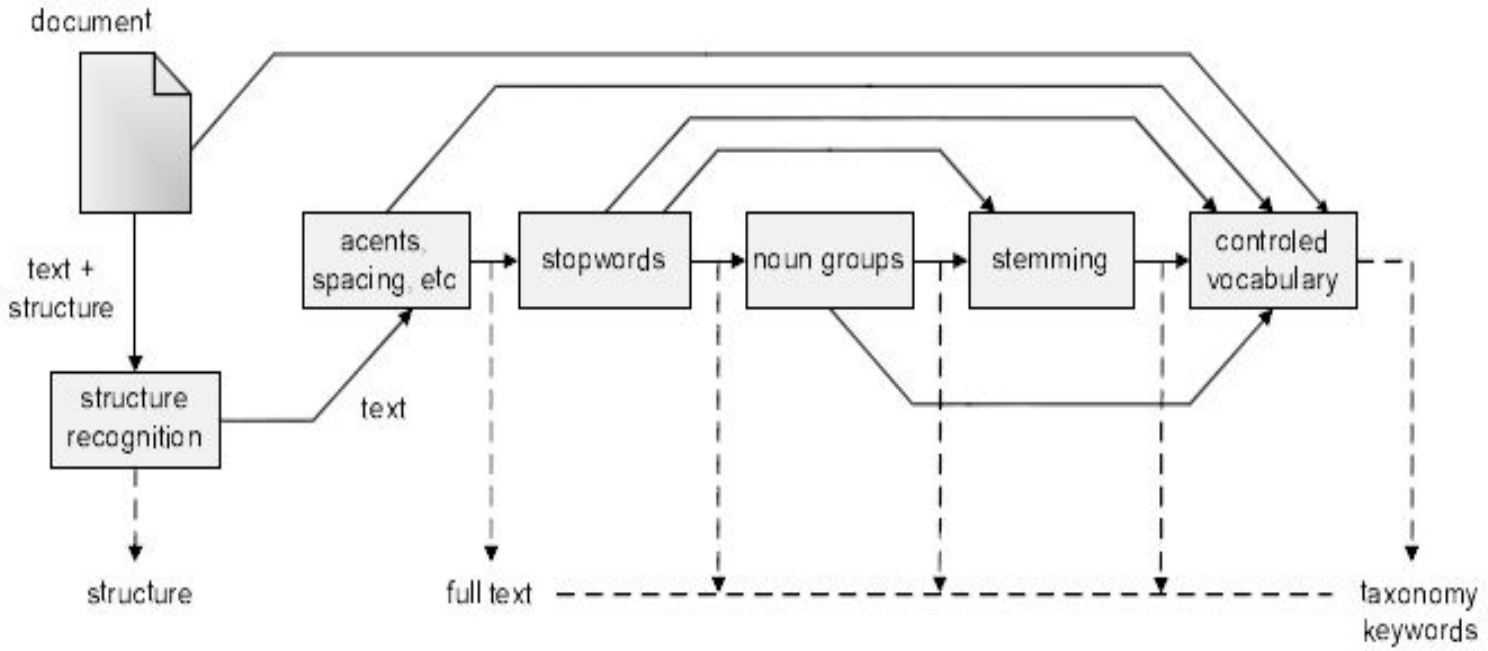
Pojęcia podstawowe

- Z każdym dokumentem d_j (lub pytaniem q) kojarzy się jednoznaczny komponent koniunktywny termu $c(d_j)$ (lub zapytania $c(q)$)
- Pojawienie się termu k_i w dokumencie d_j ustala relację między nimi, która jest opisana jako częstotliwość $f_{i,j}$ termu k_i w dokumencie d_j

$$\begin{array}{c} \\ k_1 \\ k_2 \\ k_3 \end{array} \begin{array}{c} \begin{array}{cc} d_1 & d_2 \end{array} \\ \begin{bmatrix} f_{1,1} & f_{1,2} \\ f_{2,1} & f_{2,2} \\ f_{3,1} & f_{3,2} \end{bmatrix} \end{array}$$

Pojęcia podstawowe

- Struktura logiczna dokumentu



Model boolowski

- Zapytania są specyfikowane jako wyrażenia boolowskie
  - intuicyjna i dokładna semantyka
  - zachowanie formalizmu
  - przykładowe zapytanie:

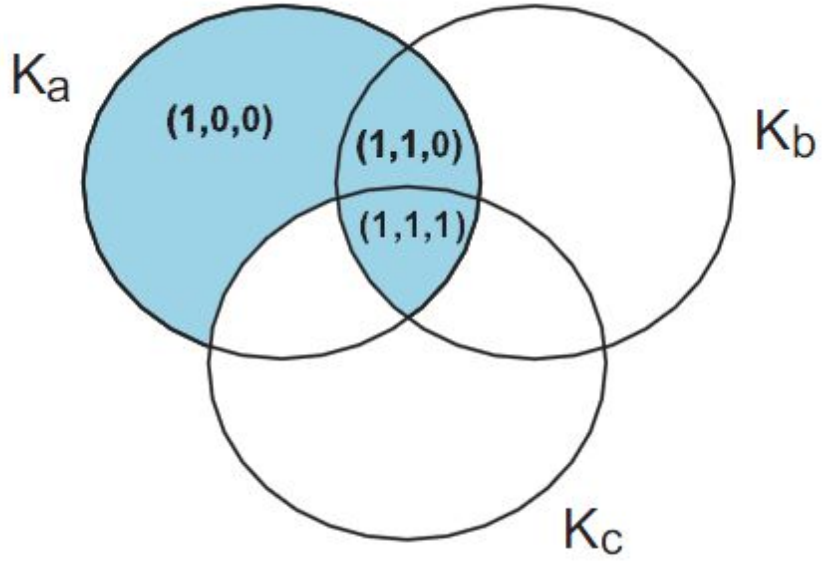
$$q = k_a \wedge (k_b \vee \neg k_c), \quad V = \{k_a, k_b, k_c\},$$

gdzie *V* oznacza słownik

- Stablicowane zależności term-dokument są binarne
  - $w_{ij} \in \{0, 1\}$: waga związana z parą (k_i, d_j)
  - $w_{iq} \in \{0, 1\}$: waga związana z parą (k_i, q)

- Koniunktywny komponent termów spełniający zapytanie *q* nazywa się koniunktywnym komponentem zapytania $c(q)$

Model boolowski



- Zapytanie *q* można zapisać w postaci dysjunktywnej $q_{DNF} = (1,1,1) \cup (1,1,0) \cup (1,0,0)$
- Jeżeli słownik $V = \{k_a, k_b, k_c, k_d\}$ a dokument d_j zawiera 3 pierwsze termy $c(d_j) = (1,1,1,0)$ to zapytanie *q* można także przedstawić w formie dysjunktywnej

Model boolowski

$$\begin{aligned} q_{DNF} = {} & (1,1,1,0) \vee (1,1,1,1) \vee \\ & (1,1,0,0) \vee (1,1,0,1) \vee \\ & (1,0,0,0) \vee (1,0,0,1) \end{aligned}$$

- Podobieństwo dokumentu d_j do zapytania q można zdefiniować jako:

$$sim(d_j, q) = \begin{cases} 1 & if\ \exists c(q) \mid c(q) = c(d_j) \\ 0 & otherwise \end{cases}$$

- Model boolowski określa czy każdy dokument jest istotny lub nie dla zapytania, bez ustalania poziomu tej istotności

Model boolowski

- Dokumenty nie podlegają rankingowi
- Zapytanie musi być tłumaczone na wyrażenie boolowskie
- W praktyce model często uwzględnia za dużo lub za mało dokumentów

Wagi termów

- Termy w dokumentach nie są jednakowo użyteczne w rozróżnianiu ich treści
  - Np. słowo występujące we wszystkich dokumentach jest bezużyteczne dla wyszukiwania

- Z termem k_i w dokumecie d_j kojarzy się wagę $w_{i,j} > 0$ (=0) gdy term pojawia się (nie pojawia) w tym dokumencie

- Waga $w_{i,j}$ wyraża ważność termu k_i dla opisu zawartości dokumentu d_j; jest istotna dla wyznaczenia rankingu dokumentu

Wagi TF-IDF

- Do ustalenia wag termów w dokumentach stosuje się formułę TF-IDF (term frequency – inverse document frequency)
- **Odwrotna częstotliwość dokumentu dla** termu k_i

$$idf_i = \log \frac{N}{n_i},$$

gdzie n_i – ilość dokumentów z termem k_i, N – *wielkość kolekcji dokumentów*

- **Częstotliwość termu** k_i dla całej kolekcji dokumentów

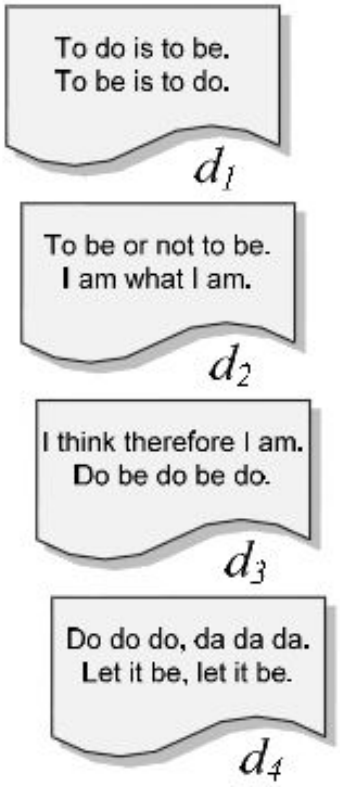
$$tf_i = 1 + \log \sum_{j=1}^{N} f_{i,j},$$

log –logarytm o podstawie 2

Wagi TF-IDF

- wagi tf-idf

$$w_{i,j} = \begin{cases} (1+\log f_{i,j}) \times \log \dfrac{N}{n_i} & if \ f_{i,j} > 0 \\ 0 & otherwise \end{cases},$$



		d_1	d_2	d_3	d_4
1	to	3	2	-	-
2	do	0.830	-	1.073	1.073
3	is	4	-	-	-
4	be	-	-	-	-
5	or	-	2	-	-
6	not	-	2	-	-
7	I	-	2	2	-
8	am	-	2	1	-
9	what	-	2	-	-
10	think	-	-	2	-
11	therefore	-	-	2	-
12	da	-	-	-	5.170
13	let	-	-	-	4
14	it	-	-	-	4

Warianty TF-IDF

weighting scheme	document term weight	query term weight
1	$f_{i,j} * \log \frac{N}{n_i}$	$(0.5 + 0.5\frac{f_{i,q}}{max_i \ f_{i,q}}) * \log \frac{N}{n_i}$
2	$1 + \log f_{i,j}$	$\log(1 + \frac{N}{n_i})$
3	$(1 + \log f_{i,j}) * \log \frac{N}{n_i}$	$(1 + \log f_{i,q}) * \log \frac{N}{n_i}$

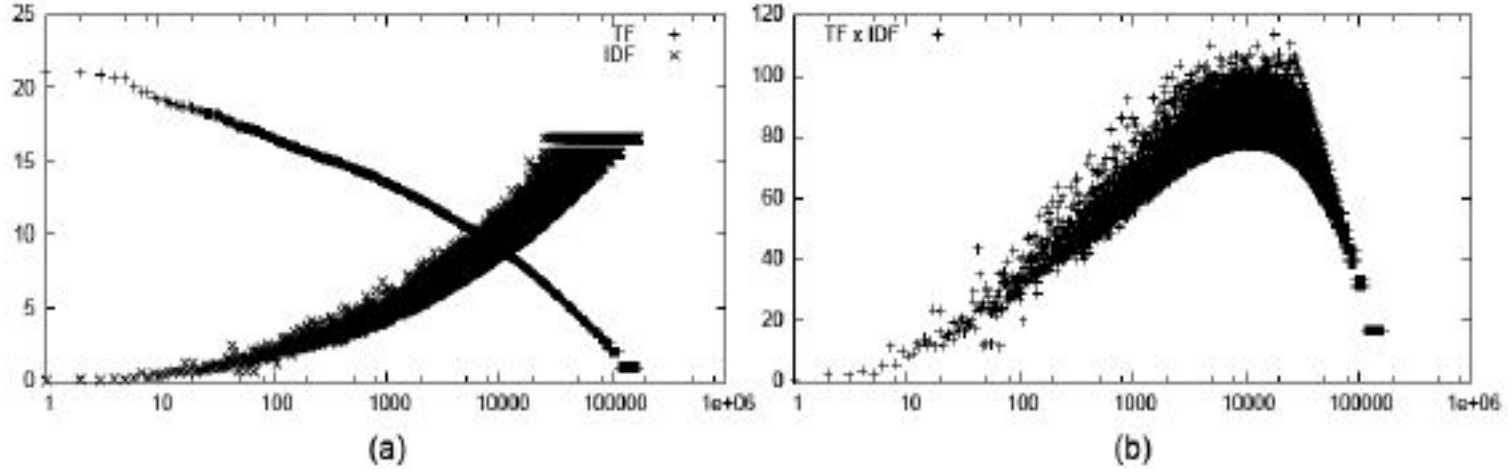
- Jeśli n_i jest częstotliwością dokumentu dla termu k_i, $n(r)$ – częstotliwością rzędu r w porządku malejącym częstotliwości to r jest rangą termu k_i oraz

$$n(r) = Nr^{-\alpha},$$

dla pewnej empirycznej stałej α (np. $\alpha = 1$).

Warianty TF-IDF

- Przykładowe wykresy *tf-idf* (tf dla kolekcji dokumentów) względem rangi termów



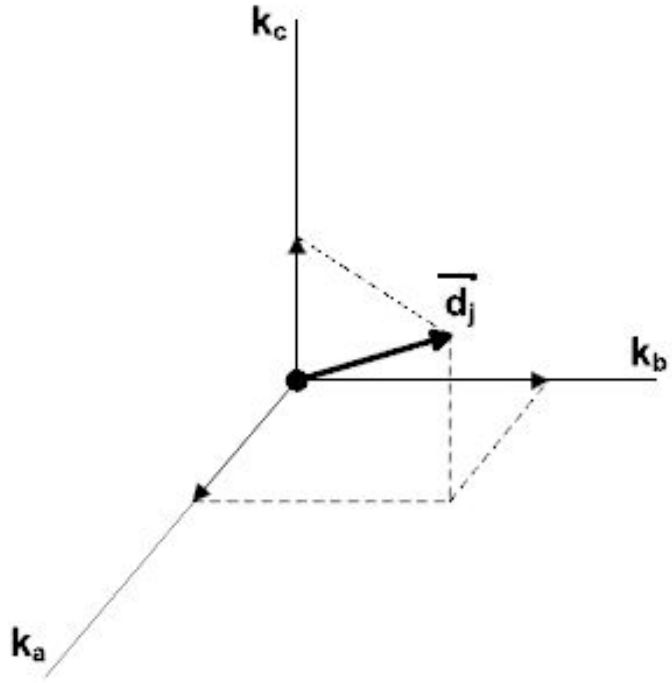
- Termy dla pośrednich wartości *idf* oraz *tf* dają maksimum wag *tf-idf* i są najbardziej odpowiednie dla rankingu

Normalizacja długości dokumentu

- Dokumenty posiadają różne długości; dłuższe z nich, a niekoniecznie bardziej istotne, mają większą szansę być wyszukane w danym zapytaniu
- Aby usunąć ten niepożądany efekt dzieli się rangę każdego dokumentu przez jego długość; ten proces nazywamy **normalizacją długości dokumentu**
- Metody normalizacji:
  - **Rozmiar w bajtach:** każdy dokument traktuje się jako strumień bajtów,
  - **Liczba słów:** dokument jest traktowany jako pojedynczy łańcuch słów do zliczenia
  - **Normy wektorowe:** dokumenty są reprezentowane jako wektory termów ważonych

Normalizacja długości dokumentu

- Każdy term w zbiorze dokumentów jest związany z wersorem $\mathbf{k_i}$ w przestrzeni t-wymiarowej
- Term k_i w dokumencie d_j jest skojarzony ze składową $w_{i,j} \times k_i$ wektora tego dokumentu



Normalizacja długości dokumentu

- wektor dokumentu:

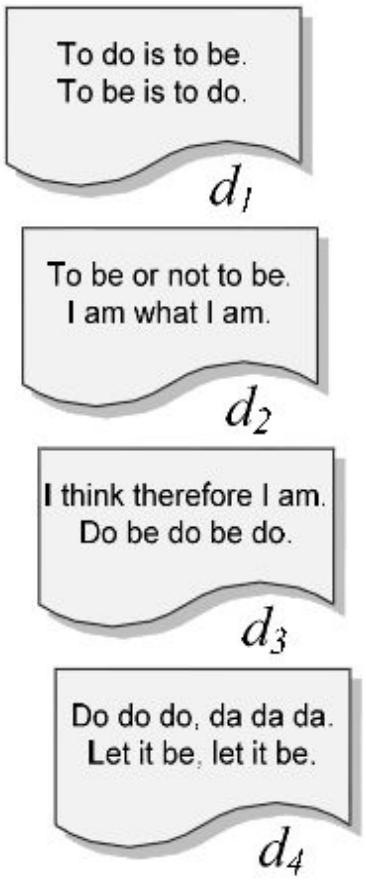
$$\mathrm{d}_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j}),$$

- długość dokumentu:

$$|\mathrm{d}_j| = \sqrt{\sum_{i}^{t} w_{i,j}^2},$$

Normalizacja długości dokumentu

- Trzy warianty długości dokumentu w przykładowej kolekcji:



	d_1	d_2	d_3	d_4
size in bytes	34	37	41	43
number of words	10	11	10	12
vector norm	5.068	4.899	3.762	7.738

Model wektorowy

- Model wektorowy uwzględnia nie tylko wystąpienie dopasowania, ale także stopień dopasowania dokumentów do zapytania poprzez wagi poszczególnych termów

- Dokumenty są uszeregowane w porządku malejącym stopnia zgodności z zapytaniem

- W modelu wektorowym:
  - Wagi $w_{i,j}$ związane z parami (k_i, d_j) są dodatnie i niebinarne
  - Termy indeksujące występują niezależnie od siebie (z założenia)
  - Termy są reprezentowane przez wersory w przestrzeni t-wymiarowej.

Model wektorowy

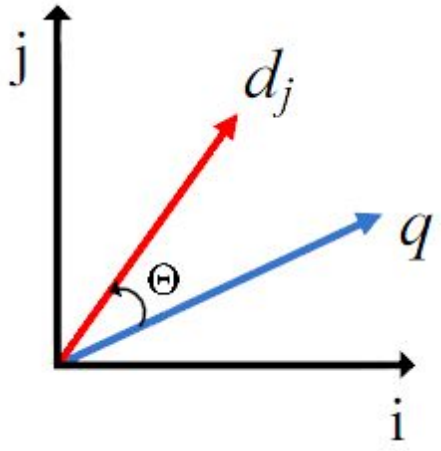
- W modelu wektorowym: reprezentacje dokumentu d_j oraz zapytania q są wektorami:

$$\mathrm{d}_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j}), \quad \mathrm{q} = (w_{1,q}, w_{2,q}, \ldots, w_{t,q}),$$

- Podobieństwo dokumentu d_j oraz zapytania q:

$$\cos(\theta) = \frac{\mathbf{d_j} \bullet \mathbf{q}}{|\mathbf{d_j}| \times |\mathbf{q}|}$$

$$sim(d_j, q) = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}$$



Model wektorowy

- Wagi w modelu wektorowym są typu tf-idf

$$w_{i,q} = (1 + \log f_{i,q}) \times \log \frac{N}{n_i},$$

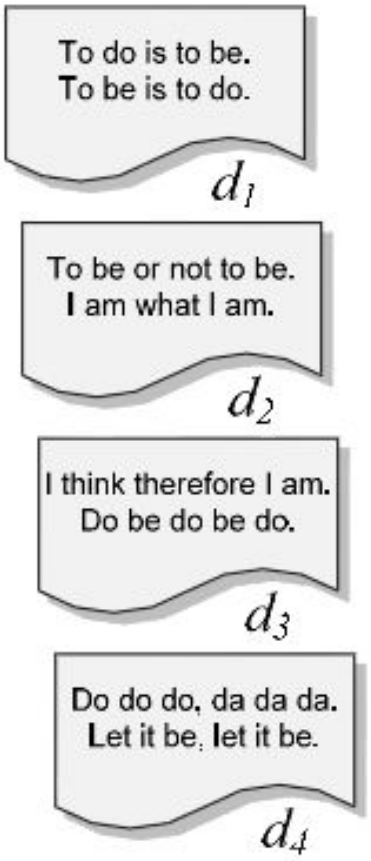
$$w_{i,j} = (1 + \log f_{i,j}) \times \log \frac{N}{n_i},$$

gdzie n_i – ilość dokumentów zawierających term k_i, N- ilość wszystkich dokumentów

- Te równania stosuje się dla termów o częstotliwości większej od 0
- Jeżeli częstotliwość termu jest zerowa odpowiednie wagi są zerowe

Model wektorowy

- Rangi przykładowych dokumentów dla zapytania „*to do*”, w modelu wektorowym z wagami tf-idf



doc	rank computation	rank
d_1	$\frac{1*3+0.415*0.830}{5.068}$	0.660
d_2	$\frac{1*2+0.415*0}{4.899}$	0.408
d_3	$\frac{1*0+0.415*1.073}{3.762}$	0.118
d_4	$\frac{1*0+0.415*1.073}{7.738}$	0.058

Model wektorowy

- Zalety modelu wektorowego:
  - wprowadza wagi termów,
  - częściowe dopasowanie pozwala na przybliżone wyszukiwanie,
  - cosinusowa funkcja rankingu sortuje dokumenty wg. podobieństwa do zapytania,
  - normalizacja długości dokumentu wbudowana w ranking.

- Wady:
  - model zakłada niezależność termów

Model probabilistyczny

- Dla danego zapytania istnieje **idealny zbiór odpowiedzi** na to pytanie**,**
- Na podstawie tego zbioru ustala się istotne dokumenty
- Zapytanie traktuje się jako specyfikację właściwości tego idealnego zbioru
- Początkowy zbiór dokumentów wybiera się dowolnie,
- Po przejrzeniu przez użytkownika 10-20 pierwszych dokumentów korygowany jest zbiór idealny
- Powtarzanie powyższego procesu doskonali zbiór idealny

Ranking probabilistyczny

- Model probabilistyczny:
  - Estymuje prawdopodobieństwo, że dokument jest istotny dla zapytania użytkownika,
  - Zakłada, że to prawdopodobieństwo zależy jedynie od zapytania i reprezentacji dokumentów,
  - Idealny zbiór odpowiedzi *R*, maksymalizuje prawdopodobieństwo istotności dokumentu
- Podobieństwo dokumentu do zapytania:

$$\text{sim}(d_j, q) = \frac{P(R \mid \mathbf{d}_j, q)}{P(\overline{R} \mid \mathbf{d}_j, q)},$$

gdzie *R* – zbiór dokumentów istotnych dla zapytania *q*,

$\overline{R}$ – zbiór dokumentów nieistotnych

Ranking

- Ze wzoru Bayes'a:

$$\mathrm{sim}(d_j,q)=\frac{P(\mathbf{d}_j \mid R,q)\times P(R,q)}{P(\mathbf{d}_j \mid \overline{R},q)\times P(\overline{R},q)} \sim \frac{P(\mathbf{d}_j \mid R,q)}{P(\mathbf{d}_j \mid \overline{R},q)},$$

gdzie:

- $P(R,q)$ – prawdopodobieństwo, że dokument losowo wybrany z kolekcji jest istotny dla zapytania q,
- $P(\mathbf{d}_j \mid R,q)$ – prawdopodobieństwo wybrania dokumentu d_j ze zbioru R,

- Niech $p_{iR} = P(k_i \mid R,q),\ \ q_{iR} = P(k_i \mid \overline{R},q),$

Ranking

- Po zlogarytmowaniu wzoru dla *sim* i przy założeniu

$$\forall k_i \notin q, \quad p_{iR} = q_{iR},$$

- uzyskuje się

$$\mathrm{sim}(d_j, q) \sim \sum_{k_i \in q \wedge k_i \in d_j} \log\left(\frac{p_{iR}}{1 - p_{iR}}\right) + \log\left(\frac{1 - q_{iR}}{q_{iR}}\right).$$

- Przyjmijmy:
  - N – liczba dokumentów w kolekcji,
  - R – liczba dokumentów istotnych dla pytania,
  - n_i – liczba dokumentów z termem k_i,
  - r_i – liczba dokumentów istotnych z termem k_i.

Tablica kontyngencji

- Na podstawie powyższych zmiennych można sformułować tablicę kontyngencji

	relevant	non-relevant	all docs
docs that contain k_i	r_i	$n_i - r_i$	n_i
docs that do not contain k_i	$R - r_i$	$N - n_i - (R - r_i)$	$N - n_i$
all docs	R	$N - R$	N

- Jeśli informacje z tablicy są znane dla zadanego pytania, można zapisać

$$p_{iR} = \frac{r_i}{R}, \quad q_{iR} = \frac{n_i - r_i}{N - R},$$

Formuła rankingu

- Wówczas równanie dla obliczenia rankingu może być zapisane jako:

$$\mathrm{sim}(d_j, q) \sim \sum_{k_i \in [q, d_j]} \log\left(\frac{r_i}{R - r_i} \times \frac{N - n_i - R + r_i}{n_i - r_i} \right),$$

- Dla małych wartości r_i dodajemy 0.5 do każdego termu we wzorze powyżej , przez co uzyskuje się równanie rankingu Robertsona-Sparcka Jonesa

$$\mathrm{sim}(d_j, q) \sim \sum_{k_i \in [q, d_j]} \log\left(\frac{r_i + 0.5}{R - r_i + 0.5} \times \frac{N - n_i - R + r_i + 0.5}{n_i - r_i + 0.5} \right),$$

- Równanie powyżej wymaga znajomości początkowych przybliżeń r_i oraz R

Estymacja r_i i R

- Jedna możliwość to przyjęcie wartości $R = r_i = 0$,
- Druga możliwość to estymacja R i r_i przez wstępne wyszukiwanie z 10-20 dokumentów i ponowne wykonanie zapytania dla estymowanych wartości.

- Można wziąć równanie $sim(d_j,q) \sim \sum_{k_i \in q \wedge k_i \in d_j} \log\left(\frac{p_{iR}}{1-p_{iR}}\right) + \log\left(\frac{1-q_{iR}}{q_{iR}}\right)$ i estymować p_{iR} oraz q_{iR}

  - $p_{iR} = 0.5$, $q_{iR} = n_i / N$ gdzie n_i to liczba dokumentów z termem k_i
  - powyższe przybliżenie umożliwia wyliczenie początkowego rankingu $sim(d_j,q) \sim \sum_{k_i \in q \wedge k_i \in d_j} \log\left(\frac{N-n_i}{n_i}\right)$,

Poprawa rankingu początkowego

- Ponowne przeliczenie estymat; dodajemy 0.5 aby uniknąć problemów przy $D_i = 1$ oraz $D_i = 0$:

$$p_{iR} = \frac{D_i + 0.5}{D + 1}, \quad q_{iR} = \frac{n_i - D_i + 0.5}{N - D + 1},$$

- gdzie
  - D – zbiór dokumentów wstępnie wyszukanych,
  - D_i – zbiór dokumentów zawierających term k_i.

- Powyższy proces można powtórzyć wielokrotnie

Plusy i minusy

- Zalety:
  - Dokumenty s szeregowane wg. malejącej istotności
- Wady:
  - trzeba założyć początkową wartość p_{iR},
  - brak normalizacji długości dokumentu,
  - metoda nie stosuje wag tf

Uogólniony model wektorowy

- Modele klasyczne zakładają wzajemną niezależność termów indeksujących
- W modelu wektorowym przyjmuje się

$$\forall_{i,j} \Rightarrow \mathbf{k}_i \bullet \mathbf{k}_j = 0,$$

- W modelu uogólnionym wektory wektory termów indeksujących nie muszą być ortogonalne

- Założenia:
  - $w_{i,j}$ stanowi wagę binarną skojarzoną z $[k_i, d_j]$,
  - $V=\{k_1, k_2, \ldots, k_t\}$ zbiór wszystkich termów.

Uogólniony model wektorowy

$$
\begin{array}{rcl}
 & & (k_1, k_2, k_3, \ldots, k_t) \\
m_1 & = & (0,\ 0,\ 0,\ \ldots,\ 0) \\
m_2 & = & (1,\ 0,\ 0,\ \ldots,\ 0) \\
m_3 & = & (0,\ 1,\ 0,\ \ldots,\ 0) \\
m_4 & = & (1,\ 1,\ 0,\ \ldots,\ 0) \\
 & \vdots & \\
m_{2^t} & = & (1,\ 1,\ 1,\ \ldots,\ 1)
\end{array}
$$

Minterm to term składający się z literałów połączonych logicznym symbolem koniunkcji, który dla dokładnie jednej kombinacji wejść danej funkcji przyjmuje wartość 1.

- Dla każdego dokumentu d_j istnieje minterm $m_r = c(d_j)$ zawierający tylko termy z tego dokumentu i żadnego innego
- Takie mintermy budują ortogonalne wersory $\mathbf{m_r}$ przestrzeni 2^t wymiarowej

Uogólniony model wektorowy

- Ortogonalność wektorów $\mathbf{m_r}$ nie oznacza niezależności termów indeksujących k_i, które są skorelowane w ramach wektorów $\mathbf{m_r}$.

$$on(i, m_r) = \begin{cases} 1 & gdy\ k_i \in m_r \\ 0 & gdy\ k_i \notin m_r \end{cases}$$

- Wektor związany z termem k_i oblicza się jako:

$$\mathbf{k}_i = \frac{\sum_{\forall r} on(i_m, r)\, c_{i,r}\, \mathbf{m}_r}{\sqrt{\sum_{\forall r} on(i_m, r)\, c_{i,r}^2}}, \qquad c_{i,r} = \sum_{d_j | c(d_j) = m_r} w_{i,j}$$

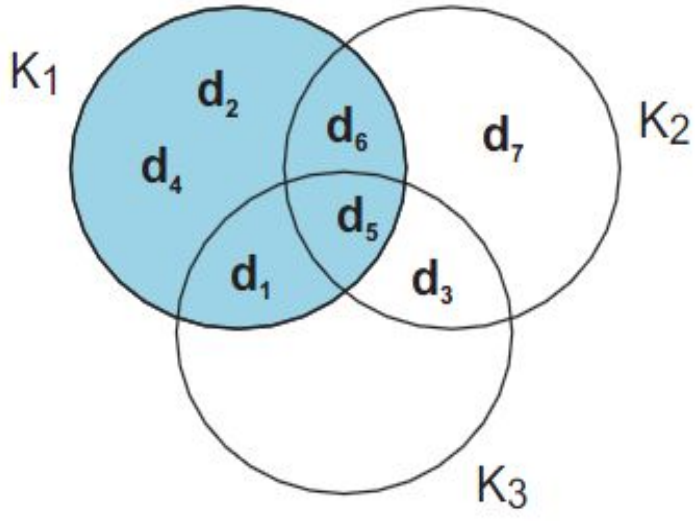
- Dla kolekcji N dokumentów tylko N mintermów (nie 2^t) uczestniczy w rankingu

Uogólniony model wektorowy

- Stopień korelacji termów k_i i k_j oblicza się jako

$$\mathbf{k}_i \bullet \mathbf{k}_j = \sum_{\forall r} on(i, m_r) \times c_{i,r} \times on(j, m_r) \times c_{j,r}$$

- Przykład:



	K_1	K_2	K_3
d_1	2	0	1
d_2	1	0	0
d_3	0	1	3
d_4	2	0	0
d_5	1	2	4
d_6	1	2	0
d_7	0	5	0
q	1	2	3

Uogólniony model wektorowy

- Obliczenie $c_{i,r}$

	K_1	K_2	K_3
d_1	2	0	1
d_2	1	0	0
d_3	0	1	3
d_4	2	0	0
d_5	1	2	4
d_6	0	2	2
d_7	0	5	0
q	1	2	3

	K_1	K_2	K_3
$d_1 = m_6$	1	0	1
$d_2 = m_2$	1	0	0
$d_3 = m_7$	0	1	1
$d_4 = m_2$	1	0	0
$d_5 = m_8$	1	1	1
$d_6 = m_7$	0	1	1
$d_7 = m_3$	0	1	0
$q = m_8$	1	1	1

	$c_{1,r}$	$c_{2,r}$	$c_{3,r}$
m_1	0	0	0
m_2	3	0	0
m_3	0	5	0
m_4	0	0	0
m_5	0	0	0
m_6	2	0	1
m_7	0	3	5
m_8	1	2	4

Uogólniony model wektorowy

- Obliczenie $\mathbf{k_i}$ i=1,2,3

$$\overrightarrow{k_1} = \frac{(3\vec{m}_2+2\vec{m}_6+\vec{m}_8)}{\sqrt{3^2+2^2+1^2}}$$

$$\overrightarrow{k_2} = \frac{(5\vec{m}_3+3\vec{m}_7+2\vec{m}_8)}{\sqrt{5+3+2}}$$

$$\overrightarrow{k_3} = \frac{(1\vec{m}_6+5\vec{m}_7+4\vec{m}_8)}{\sqrt{1+5+4}}$$

	$c_{1,r}$	$c_{2,r}$	$c_{3,r}$
m_1	0	0	0
m_2	3	0	0
m_3	0	5	0
m_4	0	0	0
m_5	0	0	0
m_6	2	0	1
m_7	0	3	5
m_8	1	2	4

Uogólniony model wektorowy

- Obliczenie wektorów dokumentów

- $\overrightarrow{d_1} = 2\overrightarrow{k_1} + \overrightarrow{k_3}$
- $\overrightarrow{d_2} = \overrightarrow{k_1}$
- $\overrightarrow{d_3} = \overrightarrow{k_2} + 3\overrightarrow{k_3}$
- $\overrightarrow{d_4} = 2\overrightarrow{k_1}$
- $\overrightarrow{d_5} = \overrightarrow{k_1} + 2\overrightarrow{k_2} + 4\overrightarrow{k_3}$
- $\overrightarrow{d_6} = 2\overrightarrow{k_2} + 2\overrightarrow{k_3}$
- $\overrightarrow{d_7} = 5\overrightarrow{k_2}$
- $\overrightarrow{q} = \overrightarrow{k_1} + 2\overrightarrow{k_2} + 3\overrightarrow{k_3}$

	K_1	K_2	K_3
d_1	2	0	1
d_2	1	0	0
d_3	0	1	3
d_4	2	0	0
d_5	1	2	4
d_6	0	2	2
d_7	0	5	0
q	1	2	3

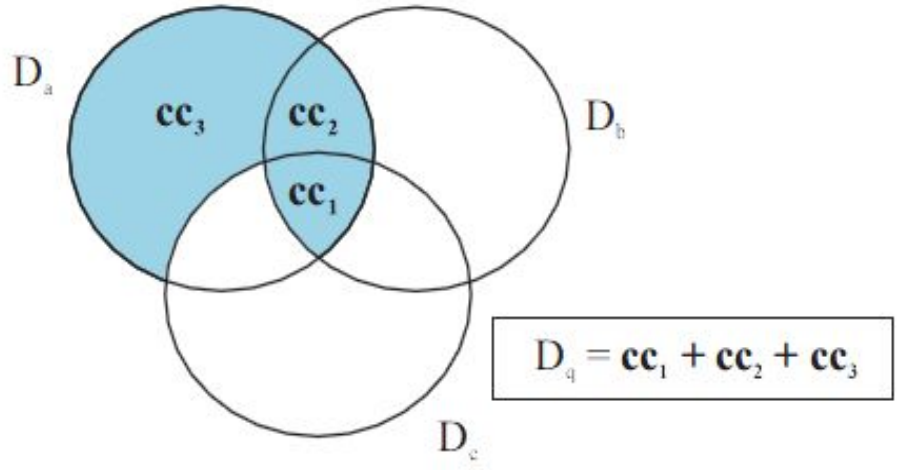
Model rozmyty

- Dopasowanie dokumentu do termów zapytania ma charakter przybliżony
- Każdy term zapytania k_i definiuje zbiór rozmyty skojarzonych z nim dokumentów,
- Każdy dokument posiada określony stopień przynależności $\mu_i \in [0,1]$ do takiego zbioru

- Taka interpretacja jest podstawą wszystkich rozmytych modeli IR

Model rozmyty Ogawa, Morita, Kobayashi

- zapytanie boolowskie dla zbioru rozmytego dokumentów $q = k_a \wedge (k_b \vee \neg k_c)$



- D_a, D_b, D_c – zbiory rozmyte dokumentów odpowiednio dla termów k_a, k_b, k_c
- cc_i, i=1,2,3 – komponenty koniunktywne
- D_q –rozmyty zbiór zapytania

Model rozmyty

- Dla zwykłych zbiorów D_i dysjunktywna postać normalna zapytania składa się z 3 koniunktywnych komponentów *cc*

$$q_{dnf} = (1, 1, 1) + (1, 1, 0) + (1, 0, 0)$$
$$= cc_1 + cc_2 + cc_3$$

- W tym modelu przyjmuje się funkcję przynależności do iloczynu zbiorów rozmytych jako iloczyn a nie minimum z tych przynależności

$$\mu_{A \cap B}(u) = \mu_A(u)\mu_B(u)$$

- Niech $\mu_{a,j}$, $\mu_{b,j}$, $\mu_{c,j}$ oznaczają stopnie przynależności documentu d_j do zbiorów rozmytych D_a, D_b, D_c.

Model rozmyty

- Wówczas:

$$
\begin{aligned}
cc_1 &= \mu_{a,j}\mu_{b,j}\mu_{c,j} \\
cc_2 &= \mu_{a,j}\mu_{b,j}(1-\mu_{c,j}) \\
cc_3 &= \mu_{a,j}(1-\mu_{b,j})(1-\mu_{c,j})
\end{aligned}
$$

- Przynależność dokumentu d_j do zapytania q:

$$
\begin{aligned}
\mu_{q,j} &= \mu_{cc_1+cc_2+cc_3,j} \\
&= 1-\prod_{i=1}^{3}(1-\mu_{cc_i,j}) \\
&= 1-(1-\mu_{a,j}\mu_{b,j}\mu_{c,j}) \times \\
&\quad (1-\mu_{a,j}\mu_{b,j}(1-\mu_{c,j})) \times (1-\mu_{a,j}(1-\mu_{b,j})(1-\mu_{c,j}))
\end{aligned}
$$

Model rozmyty

- Aby powiązać dokument d_j z termem k_i poprzez zbiór rozmyty buduje się słownik jako macierz korelacji C typu term-term

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

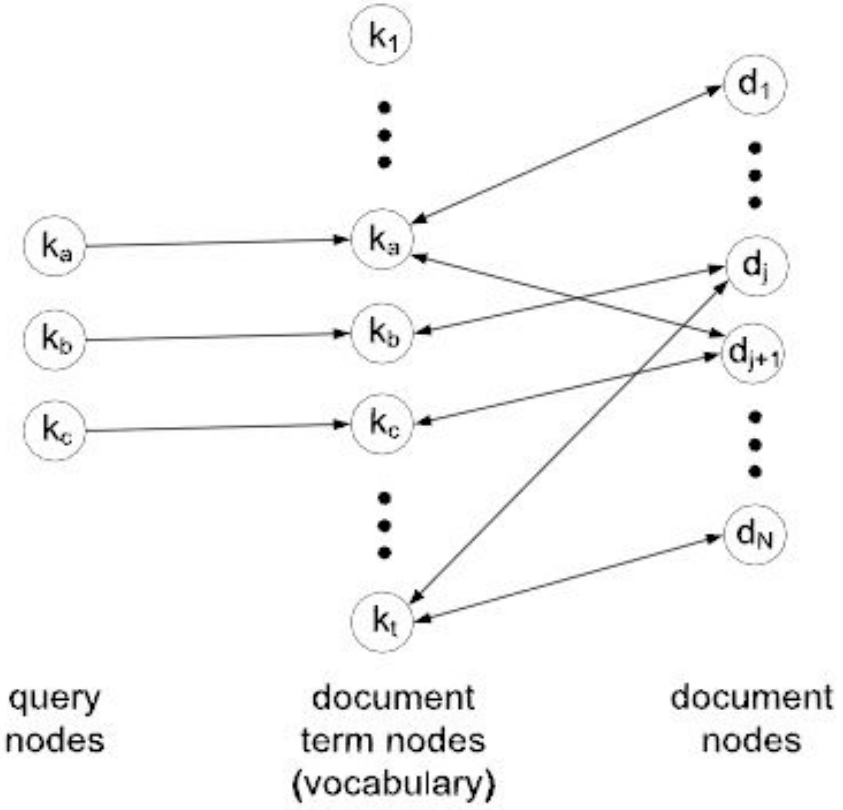
gdzie n_i – liczba dokumentów zawierających k_i, n_l – liczba dokumentów zawierających k_l, $n_{i,l}$ – liczba dokumentów zawierających zarówno k_i jak k_l

- W zbiorze rozmytym dokumentów z termem k_i dokument d_j posiada stopień uczestnictwa $\mu_{i,j}$ określony przez korelację k_i i innych termów indeksujących w dokumencie d_j

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

Model neuronowy

- Model sieci neuronowej dla wyszukiwania informacji



- Węzły dokumentów i ich termów mają wbudowane progi aktywacji

Model neuronowy

- siła sygnałów propagowanych między węzłami sieci jest wyrażona poprzez wagi połączeń synaptycznych

- Termy zapytań emitują sygnały jednostkowe
- Wagi powiązań między węzłami termów zapytań k_q i termów dokumentowych k_i. $w_{i,q} = \frac{w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,q}^2}}$,

- Wagi powiązań węzła termu dokumentowego k_i z węzłem dokumentu d_j $w_{i,j} = \frac{w_{i,j}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2}}$,

Model neuronowy

- Poziom aktywacji węzła dokumentowego d_j odpowiadający modelowi wektorowemu

$$\sum_{i=1}^{t} \overline{w}_{i,q}\,\overline{w}_{i,j} = \frac{\sum_{i=1}^{t} w_{i,q}\,w_{i,j}}{\sqrt{\sum_{i=1}^{t} w_{i,q}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,j}^2}},$$

- Nowe sygnały mogą być wymieniane między węzłami dokumentów i termów dokumentowych w procesie uczenia się sieci, przy ustawieniu określonego minimalnego progu aktywacji

Model BM25 (Best Match 25)

- BM25 powstał w wyniku serii eksperymentów nad modelami probabilistycznymi
- Do wyznaczania wag termów wykorzystuje on:
  - odwrotną częstotliwość dokumentu,
  - częstotliwość termów,
  - normalizację długości dokumentu.
- Klasyczny model probabilistyczny BM1 uwzględnia tylko pierwszą z wymienionych pozycji
- Formuła BM1 rankingu dokumentu gdy brak informacji o istotności:

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

Model BM25 (Best Match 25)

- Równanie rankingu dla modelu BM25

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

- B_{ij} stanowi współczynnik określony wzorem

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

gdzie $b \in [0,1]$ i K_1 – *stałe empiryczne, zazwyczaj* b=0.75 *oraz* K_1=1

  - b=0 – redukuje model do BM15
  - b=1 – redukuje model do BM11
- Stałe empiryczne można wyznaczyć z eksperymentów