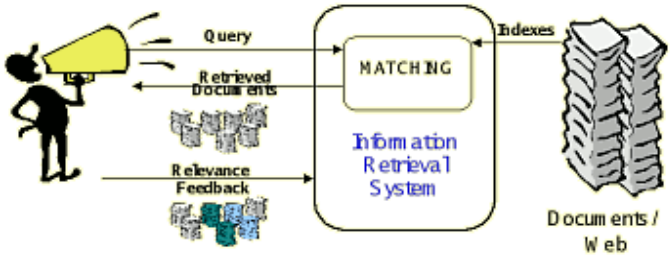


Wyszukiwanie i Przetwarzanie Informacji

Information Retrieval & Search

Irmina Masłowska

irmina.maslowska@cs.put.poznan.pl

www.cs.put.poznan.pl/imaslowska/wipi/

Podstawowe sposoby reprezentacji dokumentów tekstowych

d_1: Paul is quicker than John. Paul is quicker than George, too.
d_2: John is quicker than George.

index terms: george, john, paul, quicker

- Boolean representation (BIN - binary vectors)
  $d_1 = (1, 1, 1, 1)$
  $d_2 = (1, 1, 0, 1)$
- Bag-of words-representations (np. TF, TF-IDF)
  $d_1 = (1, 1, 2, 2)$
  $d_2 = (1, 1, 0, 1)$
- Reprezentacja pełna

Information Retrieval models

- Niech T będzie liczbą termów indeksujących (rozmiarem słownika), k_i – i-tym słowem kluczowym, D – kolekcją, czyli zbiorem wszystkich dostępnych dokumentów, Q – zbiorem zapytań
- $K=\{ k_1, k_2, ..., k_T \}$ jest zbiorem wszystkich termów indeksujących
- Z każdym słowem kluczowym k_i dokumentu d_j związana jest waga $a_{ij} > 0$ (ew. $a_{ij} \geq 0$), dla słów kluczowych niewystępujących w tekście dokumentu $a_{ij} = 0$
- Stąd każdemu dokumentowi przyporządkowany jest wektor $d_j = (a_{1j}, a_{2j}, ..., a_{Tj})$
- Niech g_i będzie funkcją, która zwraca wagę związaną ze słowem kluczowym k_i dowolnego T-wymiarowego wektora, np.: $g_i(d_j) = a_{ij}$
- $sim(q, d_j)$ jest funkcją rangującą, która przyporządkowuje wartości rzeczywiste parom (q, d_j): $q \in Q$, $d_j \in \mathrm{D}$
- Funkcja sim definiuje uporządkowanie (ranking) dokumentów względem zapytania

TF term weighting scheme

term frequency

waga termu $k_i \in K$ w dokumencie $d_j \in D$ rośnie wraz ze wzrostem liczby wystąpień tego słowa w tym dokumencie

$$a_{ij} = tf_{ij}$$

Hans Peter Luhn (1957)

TF term weighting scheme

- zwykła liczność wystąpień

  $tf_{ij} = f_{ij}$ $\quad$ f_{ij} – liczba wystąpień słowa k_i w dokumencie d_j

- częstość znormalizowana długością d_j

$$tf_{ij} = \frac{f_{ij}}{\sum_l f_{lj}}$$

- znormalizowana względem najczęstszego termu w d_j

$$tf_{ij} = \frac{f_{ij}}{\max_l f_{lj}}$$

- inne, np.:

$$tf_{ij} = \begin{cases} 1 + \log f_{ij} & \text{dla } f_{ij} > 0 \\ 0 & \text{dla } f_{ij} = 0 \end{cases}$$

0→0, 1→1, 2→1.3, 10→2, 10000→5

term frequency – inverse document frequency

waga termu $k_i \in K$ w dokumencie $d_j \in D$ rośnie wraz ze wzrostem jego liczby wystąpień w tym dokumencie

Hans Peter Luhn (1957)

waga termu $k_i \in K$ w każdym dokumencie kolekcji D maleje wraz ze wzrostem liczby dokumentów kolekcji, które zawierają term k_i

Karen Spärck Jones (1972)

$$a_{ij} = tf_{ij} \cdot idf_i$$

TF-IDF term weighting scheme

$$a_{ij} = tf_{ij} \cdot idf_i$$

$$idf_i = \log \frac{N}{N_i}$$

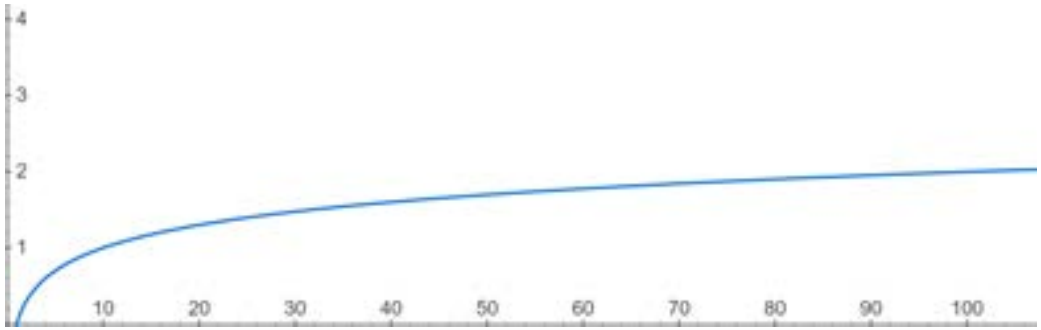
N – liczba wszystkich dokumentów kolekcji ($N=|D|$)

N_i – liczba dokumentów z D zawierających słowo k_i

TF-IDF term weighting scheme

$$a_{ij} = tf_{ij} \cdot idf_i$$

$$idf_i = \log \frac{N}{N_i}$$



N – liczba wszystkich dokumentów kolekcji ($N=|D|$)

N_i – liczba dokumentów z D zawierających słowo k_i

TF-IDF term weighting scheme

TF

dokument d_j

100 liczba wystąpień najczęstszego termu

4 wystąpienia termu $k_i = wieloryb$

$$tf_{ij} = \frac{f_{ij}}{\max_l f_{lj}}$$

IDF

10 mln dokumentów w kolekcji D

10 zawierających term $k_i = wieloryb$

(d_j i jeszcze 9 innych)

$$idf_i = \log \frac{N}{N_i}$$

TF-IDF term weighting scheme

TF

dokument d_j

100 liczba wystąpień najczęstszego termu

4 wystąpienia termu $k_i = wieloryb$

$$tf_{ij} = \frac{f_{ij}}{\max_l f_{lj}}$$

4/100 = 0.04

IDF

10 mln dokumentów w kolekcji D

10 zawierających term $k_i = wieloryb$

(d_j i jeszcze 9 innych)

$$idf_i = \log \frac{N}{N_i}$$

log(10000000/10) = 6

a_{ij} = 0,04*6 = 0.24

TF-IDF term weighting scheme

Wysoką wagę osiąga term, który w danym dokumencie występuje stosunkowo wiele razy, natomiast występuje w małym odsetku dokumentów kolekcji

10000 dokumentów, 1 zawiera słowo *wieloryb*

4/100 * log (10000 / 1) = 0.16

10000 dokumentów, 100 zawiera słowo *wieloryb*

4/100 * log (10000 / 100) = 0.08

10000 dokumentów, 10000 zawiera słowo *wieloryb*

4/100 * log (10000 / 10000) = 0

Information Retrieval models

Klasyczne modele IR

- model Boole'owski
- model wektorowy (VSM – Vector Space Model)
- model probabilistyczny (BIR – Binary Independence Model)

Nieklasyczne modele IR

- model oparty na zbiorach rozmytych (fuzzy sets)
- rozszerzony model Boole'owski
- model LSI (Latent Semantic Indexing)
- model oparty na sieciach neuronowych (NN)
- uogólniony model wektorowy (Generalized VSM)
- modele probabilistyczne (sieci Bayesowskie, belief networks, inference networks ...)

...

VSM - Vector Space Model

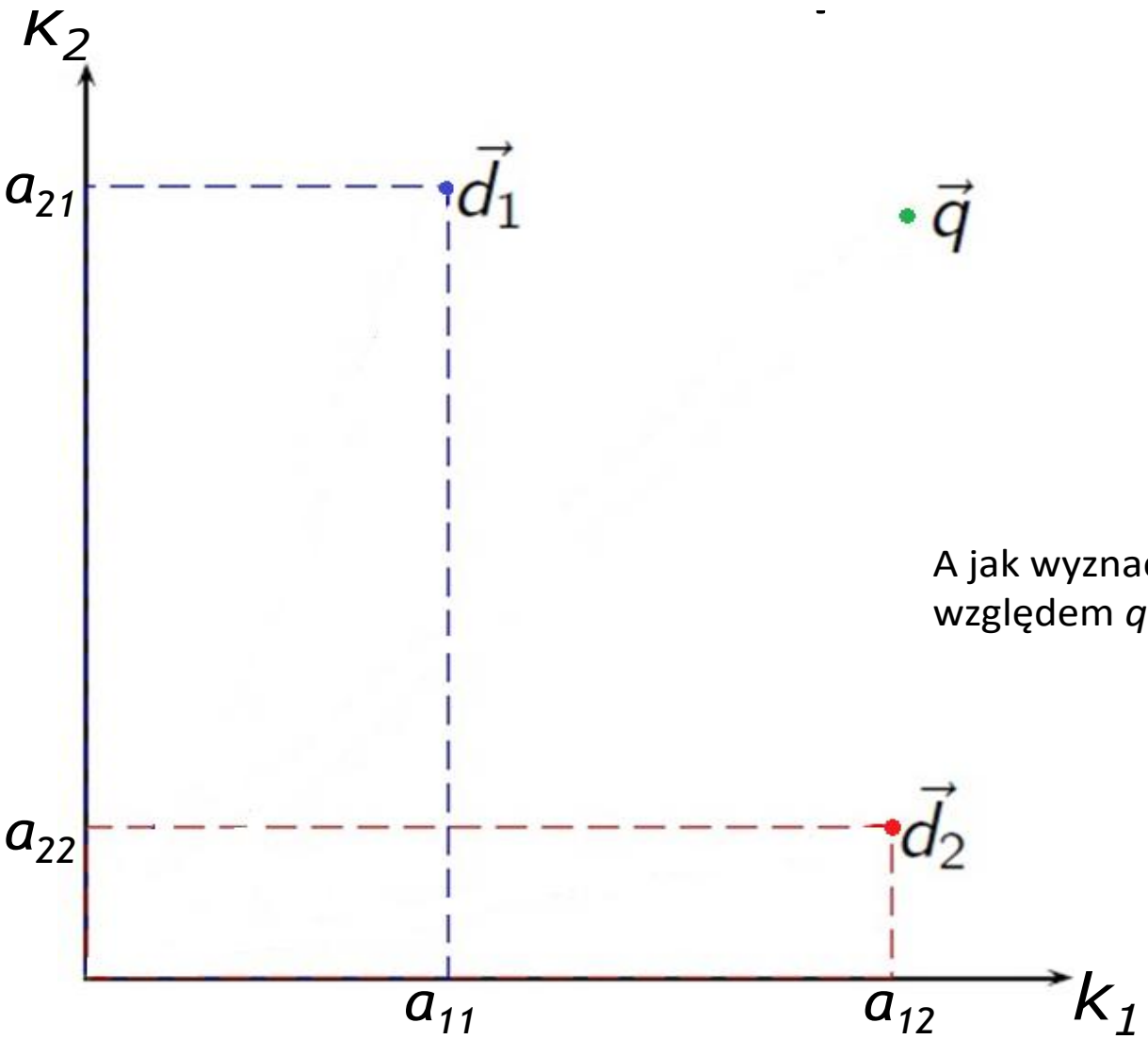
Gerard Salton, 1975

Model wektorowy VSM może bazować na reprezentacji BIN, TF, lecz najczęściej **TF-IDF**

Dokumenty i zapytania reprezentowane są w przestrzeni T-wymiarowej, gdzie T jest rozmiarem słownika (liczbą wszystkich jednostek indeksujących)

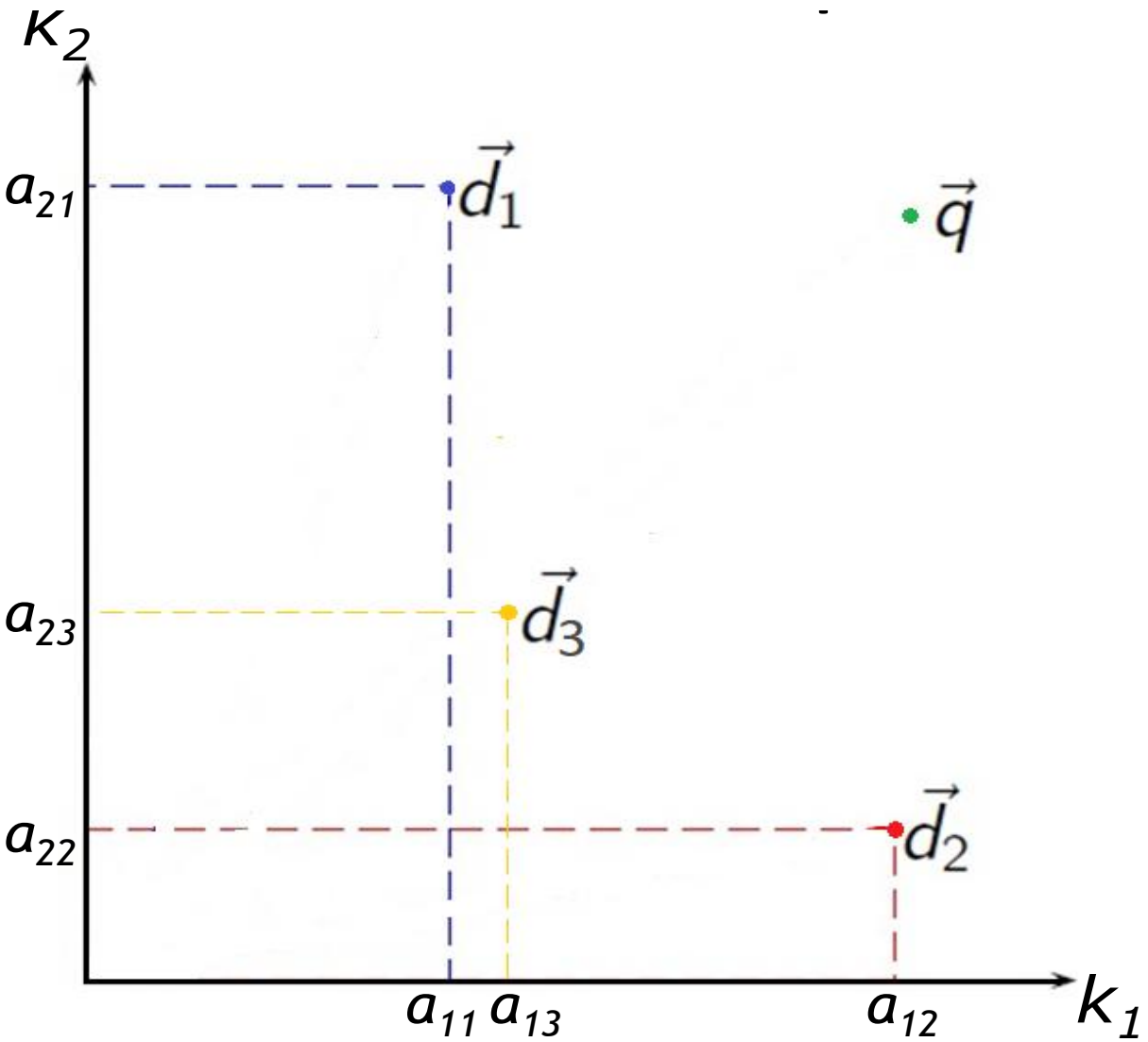
Dalej będziemy zwykle zakładać, że wagi a_{ij} poszczególnych słów kluczowych k_i dla danego dokumentu dj są wyznaczane miarą ***tf-idf***

Vector Space Model



A jak wyznaczyć ranking dokumentów względem q ?

Vector Space Model



Vector Space Model

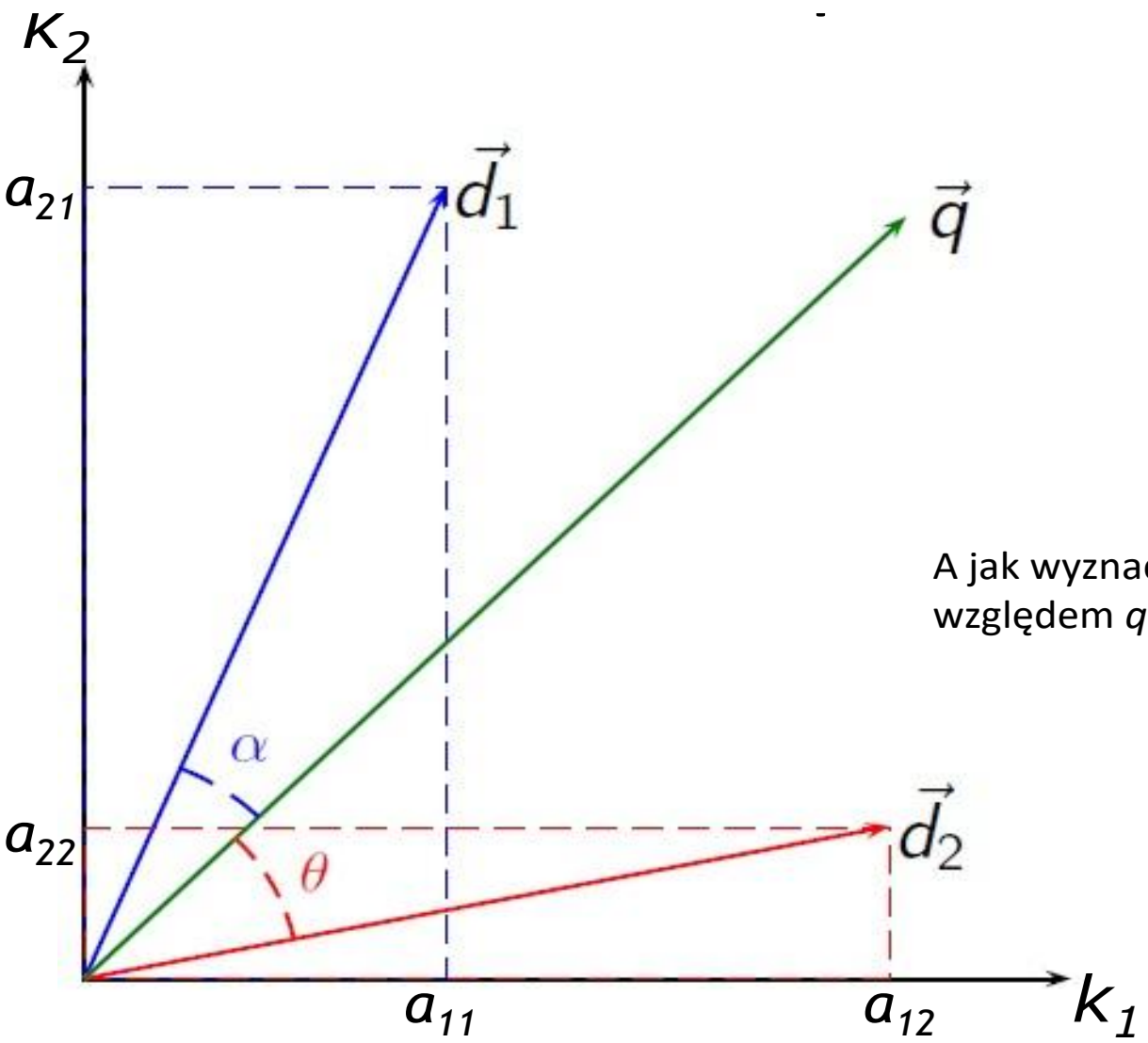
W modelu wektorowym dokumenty są *wektorami* w T-wymiarowej przestrzeni euklidesowej, gdzie osie reprezentują poszczególne termy

Każdy wektor reprezentujący dokument ma *początek* w początku układu współrzędnych, a *koniec* w punkcie o współrzędnych wyznaczonych wagami *tf-idf*

Zapytania q_i są reprezentowane analogicznie

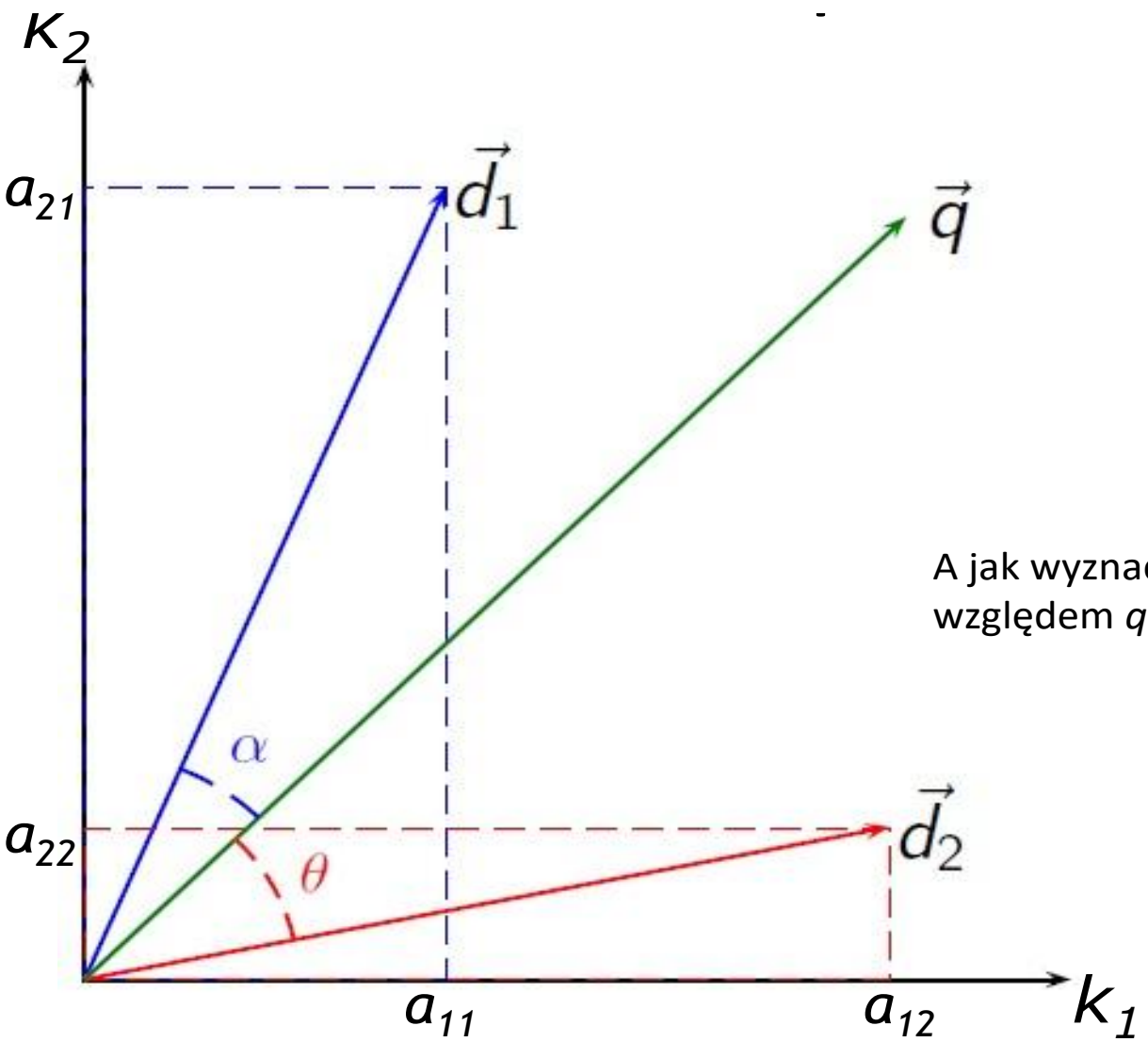
Wagi q_i mogą być binarne {0;1}, lepiej *idf lub tf-idf*

Vector Space Model



A jak wyznaczyć ranking dokumentów względem q ?

Vector Space Model



A jak wyznaczyć ranking dokumentów względem q ?

miara kosinusowa:

$$sim(d_j, q) = \frac{\vec{d}_j \bullet \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|}$$

Cosine Similarity

$$sim(d_j, q) = \frac{\vec{d}_j \bullet \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|}$$

Wektory $\vec{d}_j$ i $\vec{q}$ w T-wymiarowej przestrzeni euklidesowej

Licznik: suma iloczynów odpowiadających sobie współrzędnych

Mianownik: iloczyn długości wektorów dokumentu i zapytania (długość to pierwiastek z sumy kwadratów współrzędnych)

Miara kosinusowa nadaje się również do wyznaczania podobieństwa między dwoma dokumentami

VSM ranking

- Przedstaw zapytanie w postaci wektora *tf-idf*
- Przedstaw każdy dokument w postaci wektora *tf-idf*
- Oblicz kosinus kąta między wektorem zapytania a wektorami poszczególnych dokumentów
- Utwórz ranking dokumentów wg wartości miary kosinusowej (największe wartości na górze)
- Zwróć ustaloną liczbę najwyżej sklasyfikowanych dokumetów jako odpowiedź dla zadanego zapytania

Example Boolean Search

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0

q: Brutus AND Caesar AND NOT Calpurnia

Źródło*: An Introduction to Information Retrieval*, Cambridge Univ. P. 2009
http://nlp.stanford.edu/IR-book/

Example Boolean Search

$$\vec{b} = (1, 1, 0, 1, 0, 0)$$

$$\vec{c_s} = (1, 1, 0, 1, 1, 1)$$

$$\vec{c_l} = (0, 1, 0, 0, 0, 0)$$

$$\vec{q} = \vec{b} \wedge \vec{c_s} \wedge \neg\vec{c_l}$$

$$\vec{q} = (1, 1, 0, 1, 0, 0) \wedge (1, 1, 0, 1, 1, 1) \wedge \neg(0, 1, 0, 0, 0, 0)$$

Odpowiedź: Antony and Cleopatra, Hamlet

$$\vec{q} = (1, 0, 0, 1, 0, 0) = (1, 1, 0, 1, 0, 0) \wedge (1, 1, 0, 1, 1, 1) \wedge (1, 0, 1, 1, 1, 1)$$

Example VSM Search

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	61	0	0	0	1
Brutus	3	112	0	1	0	0
Caesar	159	145	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	56	0	0	0	0	0

	D	d_t	Idf
Antony	6	3	0.69314718
Brutus	6	3	0.69314718
Caesar	6	5	0.18232155
Calpurnia	6	1	1.79175946
Cleopatra	6	1	1.79175946

Example VSM Search

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	108.82410	42.281978	0	0	0	0.69314718
Brutus	2.0794415	77.632484	0	0.69314718	0	0
Caesar	28.989127	26.436625	0	0.36464311	0.18232155	0.18232155
Calpurnia	0	17.917594	0	0	0	0
Cleopatra	100.33853	0	0	0	0	0

q: Brutus AND Caesar AND NOT Calpurnia

Example VSM Search

q: Brutus Caesar

Ranking:
1. Hamlet
2. Antony and Cleopatra

Czy wykorzystanie IDF będzie miało wpływ
na ranking dla zapytania składającego się
z pojedynczego termu indeksującego - np. *bitcoin* ?

Czy wykorzystanie IDF będzie miało wpływ na ranking dla zapytania składającego się z pojedynczego termu indeksującego - np. *bitcoin* ?

IDF nie zmieni rankingu (mnożenie przez stałą)

Wpływ IDF objawia się przy zapytaniach składających się z 2 lub więcej termów

Przykładowo, dla zapytania *święta wielkanocne* waga IDF spowoduje, że wystąpienia słowa *wielkanocne* w dokumentach będą liczyły się dużo bardziej niż wystąpienia słowa *święta*

Vector Space Model

- Zalety modelu wektorowego
  - ☺ prostota i szybkość
  - ☺ uwzględnienie wag *tf-idf* poprawia wyniki
  - ☺ kosinusowa miara podobieństwa umożliwia uszeregowanie dokumentów zgodnie z malejącą adekwatnością (możliwość kontroli rozmiarów zbioru wyników)
  - ☺ częściowe dopasowanie umożliwia odnajdowanie dokumentów w przybliżeniu spełniających warunki zapytania
  - ☺ popularność ;)
- Wady modelu wektorowego
  - ☹ założenie o niezależności słów kluczowych

Information Retrieval models

Klasyczne modele IR

- model Boole'owski
- model wektorowy (VSM – Vector Space Model)
- model probabilistyczny (BIR – Binary Independence Model)

Nieklasyczne modele IR

- model oparty na zbiorach rozmytych (fuzzy sets)
- rozszerzony model Boole'owski
- model LSI (Latent Semantic Indexing)
- model oparty na sieciach neuronowych (NN)
- uogólniony model wektorowy (Generalized VSM)
- modele probabilistyczne (sieci Bayesowskie, belief networks, inference networks ...)

...

Binary Independence Model

Probabilistyczny model BIR bazuje na binarnej reprezentacji dokumentów i zapytań (BIN)

- Dla danego zapytania $q \in Q$ i dokumentu z kolekcji $d_j \in D$, model probabilistyczny usiłuje oszacować prawdopodobieństwo, że użytkownik uzna dokument d_j za interesujący (adekwatny)
- Prawdopodobieństwo adekwatności zależy wyłącznie od reprezentacji zapytania q i reprezentacji dokumentu d_j
- Istnieje pewien zbiór R - podzbiór dokumentów, które użytkownik preferuje jako odpowiedź na zapytanie q. Zbiór R ma maksymalizować całkowite prawdopodobieństwo adekwatności dla użytkownika. Dokumenty z R są uznawane za adekwatne, dokumenty spoza R (oznacz. R') za nieadekwatne

Funkcja rangująca modelu BIR

$$sim(d_j, q) \sim \sum_{i=1}^{T} w_{iq} \cdot a_{ij} \cdot \left(\log \frac{P(k_i \mid R)}{1 - P(k_i \mid R)} + \log \frac{1 - P(k_i \mid R')}{P(k_i \mid R')} \right)$$

w_{iq} — waga termu k_i w zapytaniu q

a_{ij} — waga termu k_i w dokumencie d_j

$P(k_i \mid R)$ — prawdopodobieństwo, że term k_i występuje w losowo wybranym dokumencie ze zbioru dokumentów interesujących R

$P(k_i \mid R')$ — prawdopodobieństwo, że term k_i występuje w losowo wybranym dokumencie ze zbioru dokumentów nieinteresujących R'

Binary Independence Model

$$sim(d_j, q) \sim \sum_{i=1}^{T} w_{iq} \cdot a_{ij} \cdot \left(\log \frac{P(k_i \mid R)}{1 - P(k_i \mid R)} + \log \frac{1 - P(k_i \mid R')}{P(k_i \mid R')} \right)$$

Początkowe szacowanie prawdopodobieństw

- prawdopodobieństwo występowania k_i w dokumencie losowo wybranym z R jest równe dla wszystkich słów kluczowych
- rozkład termów w dokumentach z R' jest taki jak w całej kolekcji dokumentów D

  $P(k_i \mid R) = 0{,}5$

  $P(k_i \mid R') = \frac{N_i}{N}$

  Dla takich wartości można otrzymać *wstępny* ranking

Binary Independence Model

Dokładniejsze szacowanie prawdopodobieństw następuje na podstawie pierwszego (wstępnego) rankingu:

- Pewna liczba dokumentów z początku wstępnego rankingu czołowych zostaje uznana za adekwatne V, reszta za nieadekwatne
- Niech V będzie podzbiorem dokumentów wybranych jako adekwatne, a V_i jego podzbiorem złożonym tylko z tych dokumentów, które zawierają term k_i, wówczas prawdopodobieństwa można oszacować następująco:

$$P(k_i \mid R) = \frac{|V_i|}{|V|}$$

$$P(k_i \mid R') = \frac{N_i - |V_i|}{N - |V|}$$

Binary Independence Model

- Zalety probabilistycznego modelu BIR
  - ☺ prostota i szybkość
  - ☺ dokumenty są porządkowane zgodnie z malejącym prawdopodobieństwem ich adekwatności (możliwość kontroli rozmiarów zbioru wyników)
- Wady modelu BIR
  - ☹ ignorowanie informacji o częstości wystąpienia słów kluczowych w dokumentach (binarne wagi)
  - ☹ konieczność zgadywania początkowego podziału zbioru dokumentów na adekwatne i nieadekwatne
  - ☹ założenie o niezależności słów kluczowych

Term Independence Assumption

- Wszystkie klasyczne modele IR bazują na założeniu, że termy indeksujące są niezależne, czyli na podstawie znajomości wagi a_{ij} przyporządkowanej parze (k_i, d_j) nie możemy nic powiedzieć o wadze a_{lj} dla pary (k_l, d_j): $i \neq l$
- Założenie o niezależności słów kluczowych jest uproszczeniem dyktowanym:
  - efektywnością i prostotą obliczeń
  - trudnością w modelowaniu związków między słowami (zależność od konkretnych kontekstów)
  - zadowalającymi wynikami mimo tego założenia

Information Retrieval models

Klasyczne modele IR

- model Boole'owski
- model wektorowy (VSM – Vector Space Model)
- model probabilistyczny (BIR – Binary Independence Model)

Nieklasyczne modele IR

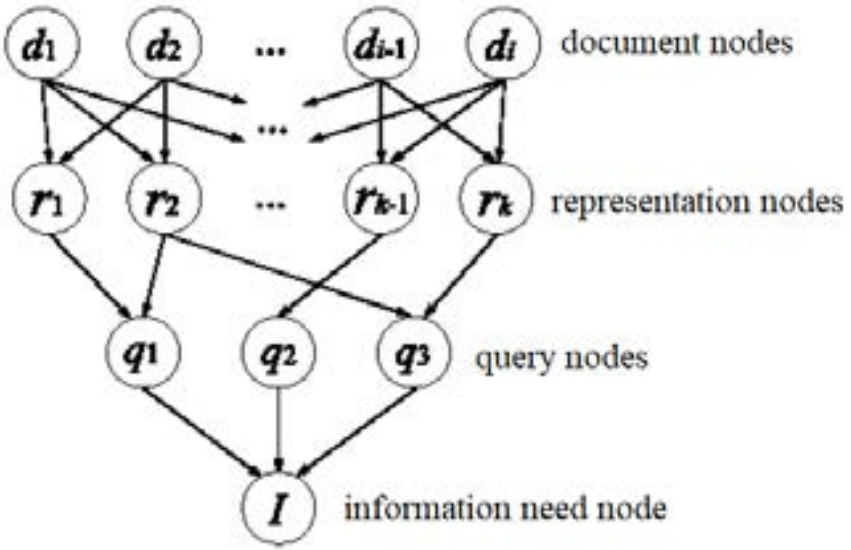
- model oparty na zbiorach rozmytych (fuzzy sets)
- rozszerzony model Boole'owski
- model LSI (Latent Semantic Indexing)
- model oparty na sieciach neuronowych (NN)
- uogólniony model wektorowy (Generalized VSM)
- modele probabilistyczne (sieci Bayesowskie, belief networks, inference networks ...)

...

Turtle and Croft (1989, 1991)

Model formalnie oparty na mechanizmie wnioskowania w sieciach bayesowskich

Sieć bayesowska to skierowany graf acykliczny kodujący prawdopodobieństwa warunkowe dla zdarzeń losowych. Wierzchołki grafu reprezentują zdarzenia, a łuki związki przyczynowe pomiędzy tymi zdarzeniami



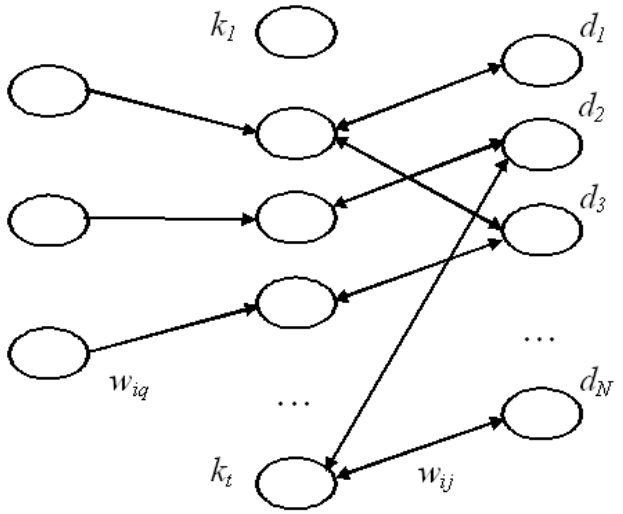
A simplified inference network

Neural Network IR model

Wilkinson and Hingston (1991)

$$w_{iq} = \frac{a_{iq}}{\sqrt{\sum_{i=1}^{T} a_{iq}^2}}$$

$$w_{ij} = \frac{a_{ij}}{\sqrt{\sum_{i=1}^{T} a_{ij}^2}}$$



zapytanie słowa kluczowe dokumenty

$$sim(d_j, q) = \sum_{i=1}^{T} w_{iq} w_{ij} = \frac{\sum_{i=1}^{T} a_{iq} a_{ij}}{\sqrt{\sum_{i=1}^{T} a_{iq}^2} \sqrt{\sum_{i=1}^{T} a_{ij}^2}}$$

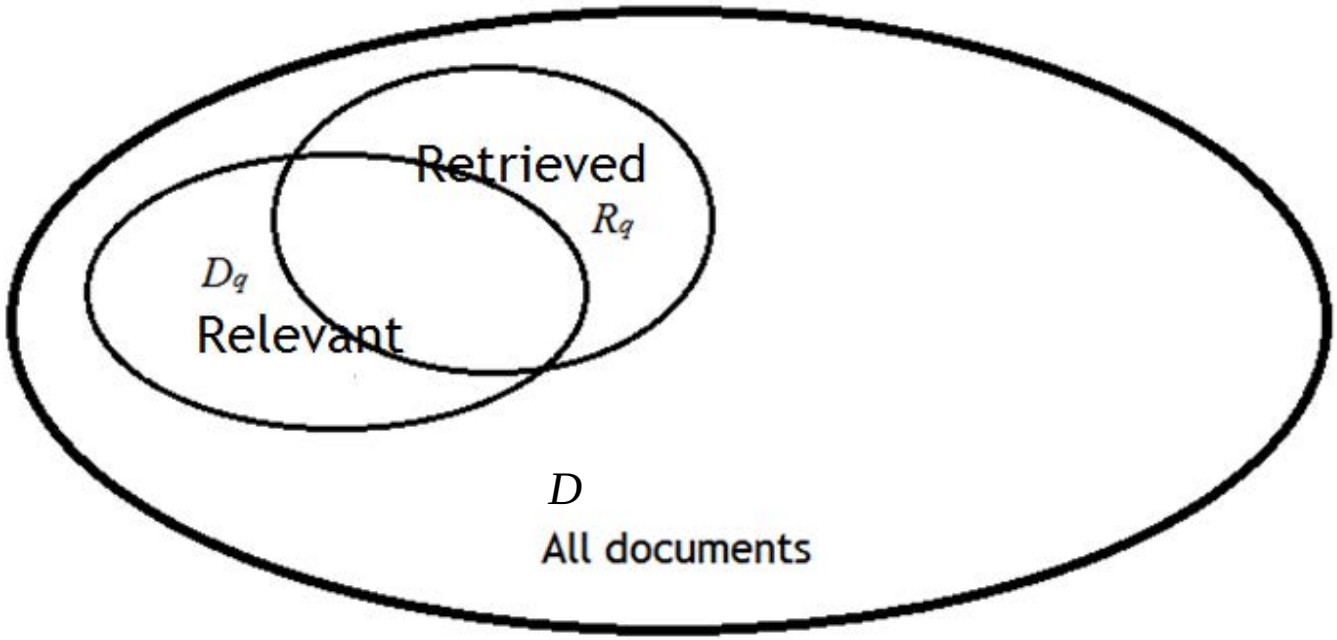
IR Evaluation Measures

Oceniając system IR należy brać pod uwagę różne aspekty jego działania:

- złożoność przetwarzania (time & space efficiency)
- jakość wyników wyszukiwania
- zadowolenie użytkowników

- rozszerzalność i adaptacyjność
- skalowalność
- ...

IR Evaluation Measures



Dla każdego zapytania q przedstawionego systemowi mamy:

- R_q – zbiór dokumentów zwróconych (Retrieved) przez system
- D_q – zbiór dokumentów adekwatnych (Relevant) obecnych w całej kolekcji dokumentów

IR Evaluation Measures

Podstawowe miary oceny jakości dopasowania odpowiedzi systemu dla zapytania q:

- miara **dokładności** (*precision*)
- miara **kompletności** (*recall*)

Podstawowe miary oceny jakości dopasowania odpowiedzi systemu dla zapytania q:

- miara **dokładności** (*precision*)
- miara **kompletności** (*recall*)

precision — odsetek wyszukanych dokumentów, które rzeczywiście są adekwatne do zapytania

$$precision = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Retrieved\}|} = \frac{|D_q \cap R_q|}{|R_q|}$$

Podstawowe miary oceny jakości dopasowania odpowiedzi systemu dla zapytania q:

- miara **dokładności** (*precision*)
- miara **kompletności** (*recall*)

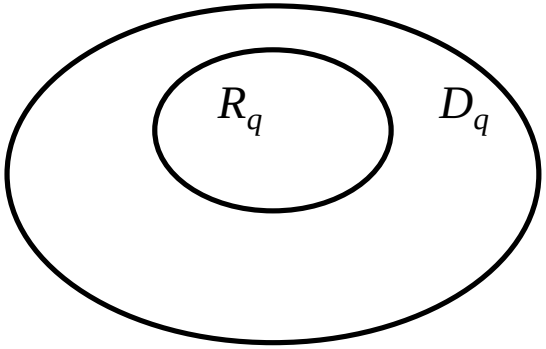
precision — odsetek wyszukanych dokumentów, które rzeczywiście są adekwatne do zapytania

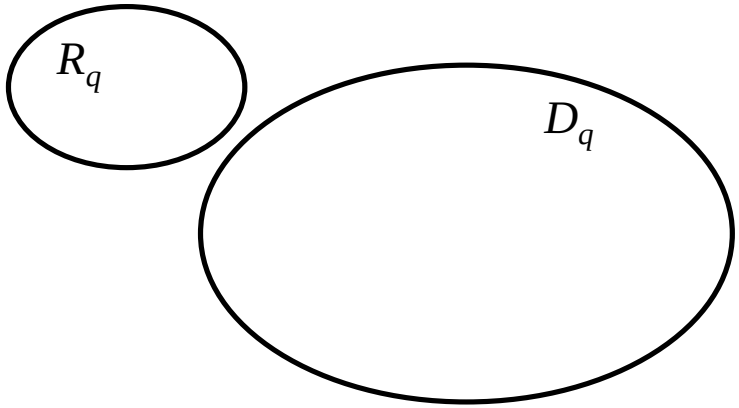
$$precision = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Retrieved\}|} = \frac{|D_q \cap R_q|}{|R_q|}$$

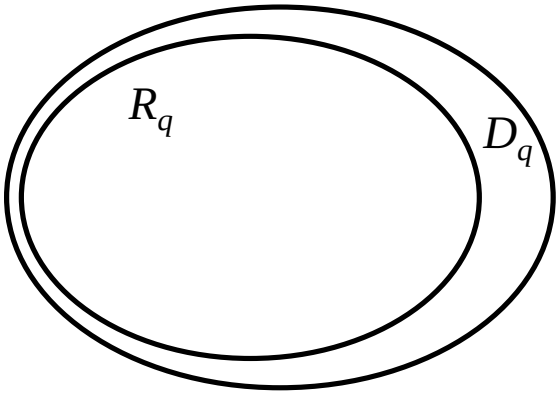
recall — odsetek dokumentów adekwatnych do zapytania, które zostały wyszukane

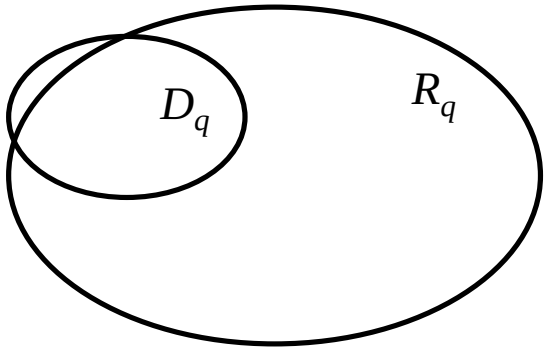
$$recall = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Relevant\}|} = \frac{|D_q \cap R_q|}{|D_q|}$$

IR Evaluation Measures









IR Evaluation Measures

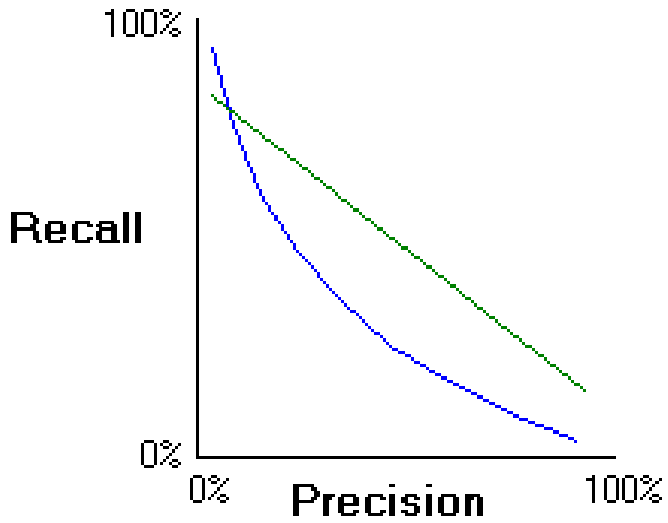
Dokładność i kompletność a konkretne zastosowania

Typowy użytkownik chciałby, aby w czołówce zwróconego rankingu były tyko dokumenty adekwatne, ale zwykle nie zechce przeglądać wszystkich dokumentów adekwatnych

Przeszukując dysk twardy, użytkownik jest zwykle zainteresowany znalezieniem wszystkich dokumentów pasujących do zapytania

A filtr antyspamowy?

Badania rzeczywistych systemów IR pokazują, że jeśli polepsza się *precision* to degraduje się *recall* jeśli polepsza się *recall* to degraduje się *precision*



F-measure – ważona średnia harmoniczna dokładności (P) i kompletności (R)

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R} \qquad \beta^2 = \frac{1-\alpha}{\alpha}$$

$$\alpha \in [0,1] \quad \beta^2 \in [0,\infty]$$

F-measure – ważona średnia harmoniczna dokładności (P) i kompletności (R)

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R} \qquad \beta^2 = \frac{1-\alpha}{\alpha}$$

$$\alpha \in [0,1] \quad \beta^2 \in [0,\infty] \qquad F_{\beta=1} = \frac{2PR}{P+R}$$

β to parametr ustalany tak, że $\beta > 1$ nacisk na kompletność, $\beta < 1$ nacisk na dokładność

Średnia harmoniczna jest bliższa minimum dwóch wartości niż średnia arytmetyczna lub geometryczna

precision at rank k ($P@k$) i **recall at rank k** ($R@k$)

Miary oceniające ranking dokumentów liczone są dla pewnej liczby k początkowych dokumentów

precision at rank k ($P@k$) i **recall at rank k** ($R@k$)

Miary oceniające ranking dokumentów liczone są dla pewnej liczby k początkowych dokumentów

Niech $r_i = 1$ gdy dokument d_i stojący na i-tej pozycji w rankingu jest adekwatny (0 w przeciwnym razie)

Niech $k >= 0$

Dokładność oraz kompletność dla k początkowych dokumentów definiujemy jako:

$$precision(k) = \frac{1}{k}\sum_{i=1}^{k} r_i \quad \text{oraz} \quad recall(k) = \frac{1}{|D_q|}\sum_{i=1}^{k} r_i$$

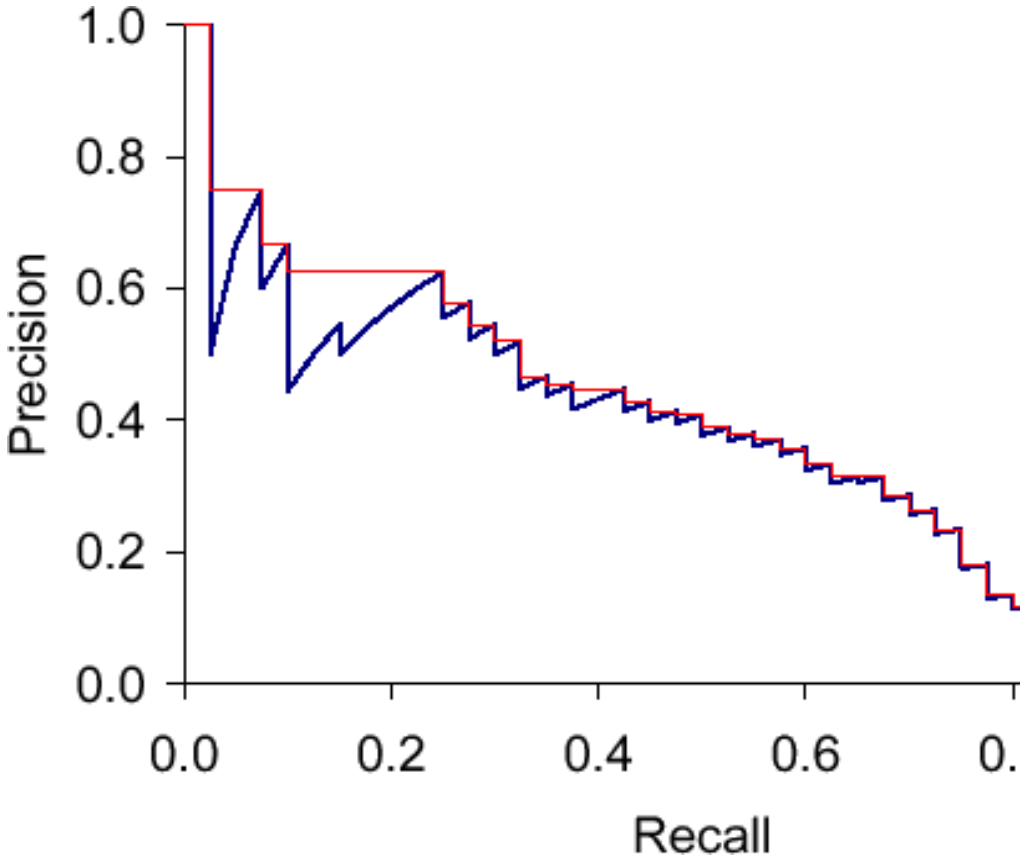
Evaluation of Ranked Retrieval Results

$$precision(k) = \frac{1}{k}\sum_{i=1}^{k} r_i \quad \text{oraz} \quad recall(k) = \frac{1}{|D_q|}\sum_{i=1}^{k} r_i$$

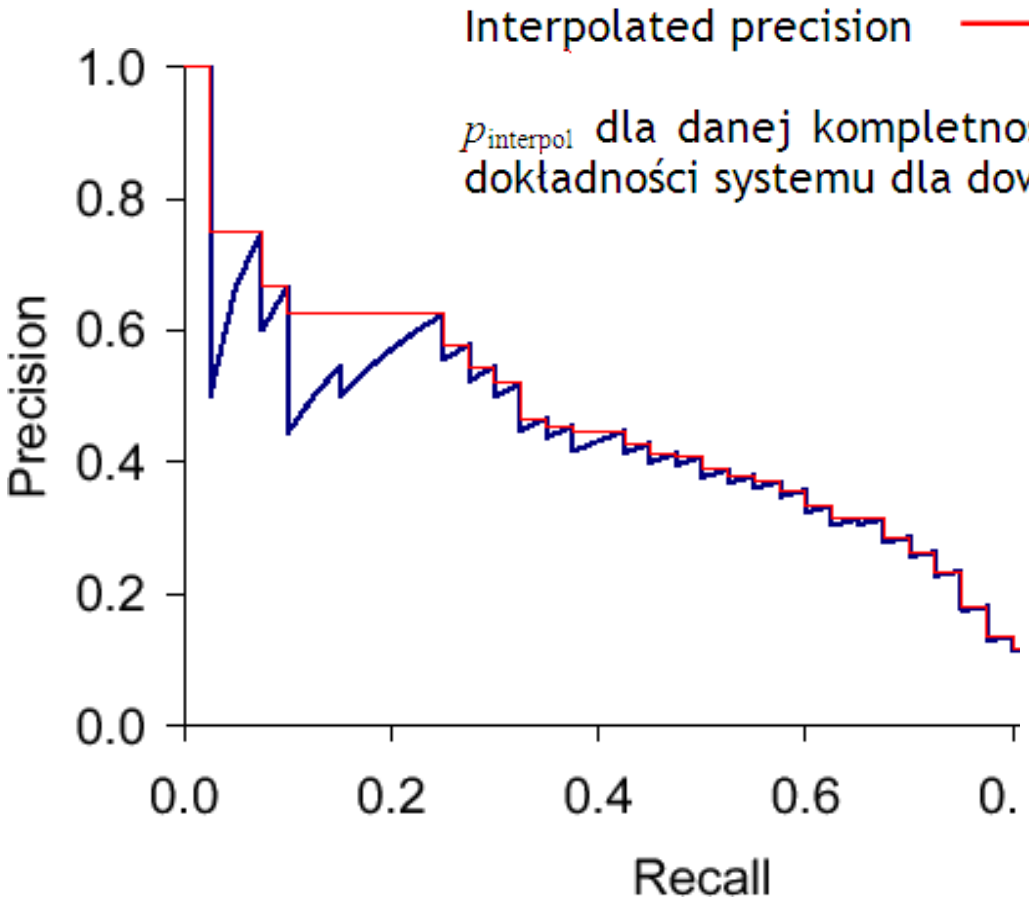
	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8
$P@k$	1/1	1/2	1/3	2/4	2/5	3/6	3/7	4/8
$R@k$	1/4	1/4	1/4	2/4	2/4	3/4	3/4	4/4*

*) cały ranking ma 8 pozycji, $|D_q|$=4

Evaluation of Ranked Retrieval Results



Evaluation of Ranked Retrieval Results



p_{interpol} dla danej kompletności r to najwyższa wartość dokładności systemu dla dowolnej kompletności $r' \geq r$

11-point interpolated average precision

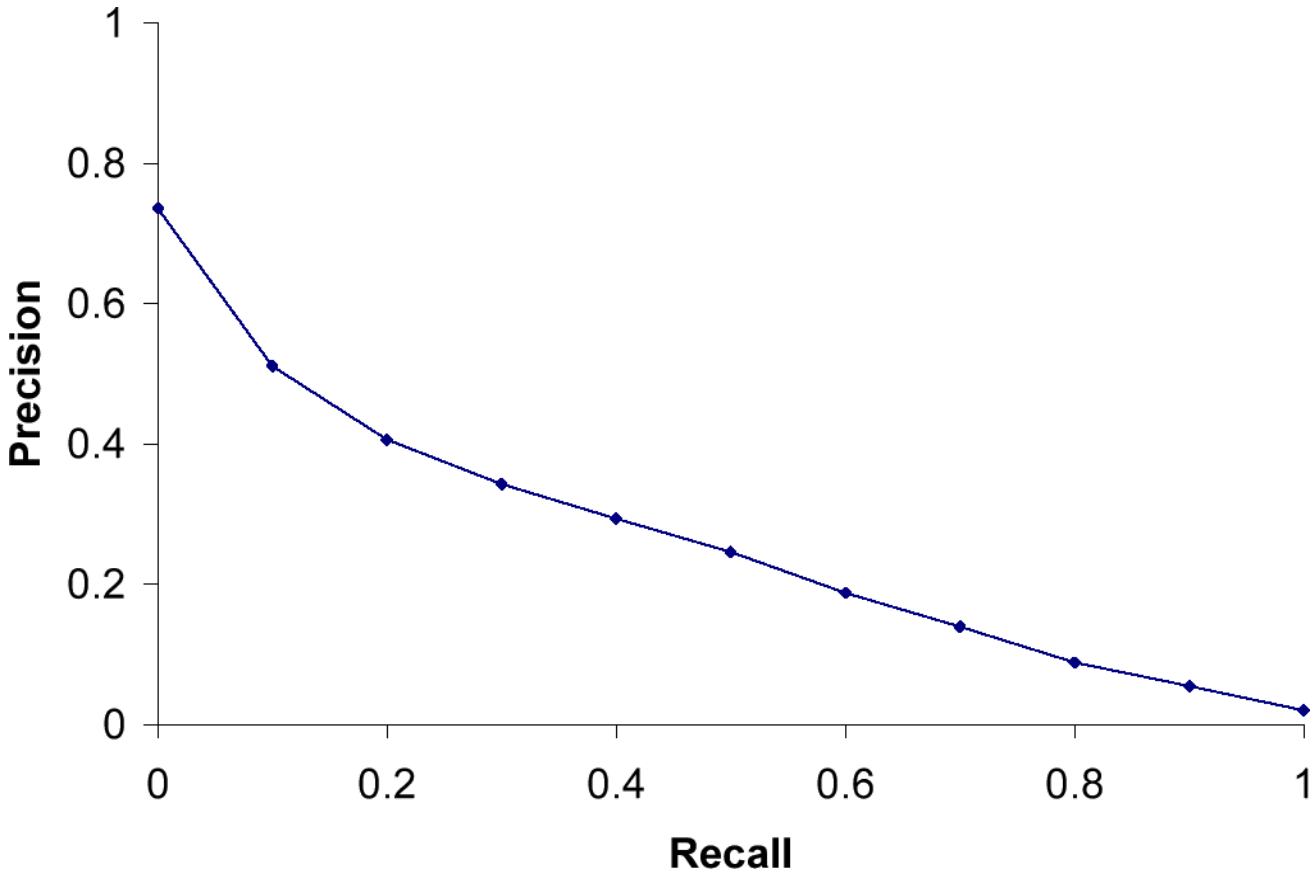
interpolowana dokładność mierzona dla 11 poziomów kompletności: 0.0, 0.1, 0.2, ..., 1.0

uśredniana po wszystkich badanych zapytaniach

miara używana m.in. dla oceny 1-8 TREC Ad Hoc Tasks

kolekcje TREC (Text Retrieval Conference)
kolekcje testowe do ewaluacji różnych zadań w ramach tematów konferencji
najpopularniejsze: Ad Hoc Track z lat 1992-1999 1,89mln docs (głównie artykuły prasowe), częściowe oceny adekwatności dla 450 zapytań;
TRECs 6–8: 528 tys. artykułów newswire i Foreign Broadcast Information Service + 150 zapytań

Evaluation of Ranked Retrieval Results



Averaged 11-point precision/recall graph across 50 queries for a representative TREC system (MAP=0.2553)

Mean average precision (MAP)

$$MAP(q) = \frac{1}{|D_q|} \sum_{j=1}^{|D_q|} \frac{j}{m_j}$$

gdzie m_j to minimalna liczba dokumentów od początku rankingu, które zawierają dokładnie j adekwatnych dokumentów

wymaga znajomości liczności zbioru dokumentów adekwatnych D_q

Mean average precision (MAP)

$$MAP(q) = \frac{1}{|D_q|} \sum_{j=1}^{|D_q|} \frac{j}{m_j}$$

gdzie m_j to minimalna liczba dokumentów od początku rankingu, które zawierają dokładnie j adekwatnych dokumentów

$$\text{MAP} = ¼\ (1+2/4+3/6+4/8) = 5/8$$

Mean average precision (MAP)

Oceny MAP obliczone dla danego systemu zwykle wykazują bardzo duże zróżnicowanie w zależności od badanego zapytania, przykładowo dla typowego systemu z TREC mogą to być wartości rozrzucone pomiędzy 0.1 and 0.7. Stąd powszechnie miarę tę uznaje się za miarodajną dla porównywania różnych systemów na danym zapytaniu

Zbiór testowy zapytań musi więc być bardzo duży i różnorodny, aby mógł służyć za podstawę oceny jakości systemu wyszukującego i stanowić bazę do porównania różnych systemów

R-precision

R-dokładność to dokładność zbioru pierwszych $|D_q|$ dokumentów w rankingu zwróconym przez system wyszukujący (czyli $P@k$ dla $k=|D_q|$)

Miara ta w praktyce silnie koreluje z miarą MAP

Obliczenie miary *R*-dokładność także wymaga znajomości zbioru dokumentów adekwatnych D_q lub choćby oszacowania jego liczności $|D_q|$

R-precision

R-dokładność to dokładność zbioru pierwszych $|D_q|$ dokumentów w rankingu zwróconym przez system wyszukujący

R-precision = 2/4 = 0.5

Other IR Evaluation Measures

Miary badające zadowolenie/frustrację użytkownika, *coverage ratio, novelty ratio, relative recall, recall effort*

- *Expected Search Length*: average number of documents which must be examined before the total number of relevant documents is reached.
- *Sliding Ratio*: sum of the relevance judgments of the documents retrieved so far divided by the sum of the relevance judgments of the documents the ideal system would have retrieved so far.
- *Novelty Ratio*: percentage of the relevant retrieved documents which were previously unknown to the user.
- *Coverage Ratio*: percentage of relevant and known documents which are retrieved.
- *Relative Recall* (aka *sought recall*): percentage of the documents the user would have liked to examine which are relevant, retrieved, and examined.
- *Recall effort*: ratio of desired to examined by the user documents.
- *Satisfaction* (and *Frustration*): sliding ratio on documents in R ($\overline{R}$) only.
- *Total*: weighted mean of satisfaction and frustration.
- *Usefulness measure*: which of two IR systems delivers more useful information to the user.
- *Average Search Length*: average number of documents examined moving down in a ranked list before the average position of a relevant document is reached.
- *NDPM*: normalized distance between user and system ranking of documents.
- *Ranked Half Life*: degree to which relevant documents are located on the top of a ranked retrieval result.
- *Relative Relevance*: degree of agreement between the types of relevance applied in a non-binary assessment context.

Exercises

I. Dla dokumentów *d1*={ala ma kota ma ala} oraz *d2*={alan kota ma kota}, oblicz współczynnik podobieństwa Jaccarda (pamiętaj, że współczynnik ten operuje na binarnej reprezentacji dokumentów)

II. Dla dokumentów z zadania powyżej przedstaw ich reprezentację za pomocą wartości TF znormalizowanych względem najczęstszego termu zdania, oraz wartości TF-IDF (zachowaj kolejność termów alfabetyczną: *T*=(ala, alan, kota, ma)). Oblicz ranking dokumentów względem zapytania *q*={ala alan} stosując kosinusową miarę podobieństwa.

III. W kolekcji 100 dokumentów, 8 jest uważanych za adekwatne (*relevant*) do zapytania *Dq*: {*d8*, *d10*, *d12*, *d20*, *d22*, *d30*, *d50*, *d88*}. W odpowiedzi na zapytanie *q* rozważany system IR zwrócił (*retrieved*) następujący ranking 12 dokumentów: *d35*, *d12*, *d8*, *d20*, *d97*, *d50*, *d10*, *d29*, *d66*, *d88*, *d22*, *d30*.

   a) oblicz miary „*Precision@k*” oraz „*Recall@k*” dla *k*=6

   b) podaj wartość miary *R-precision*

   c) przedstaw obliczenia miary *MAP*