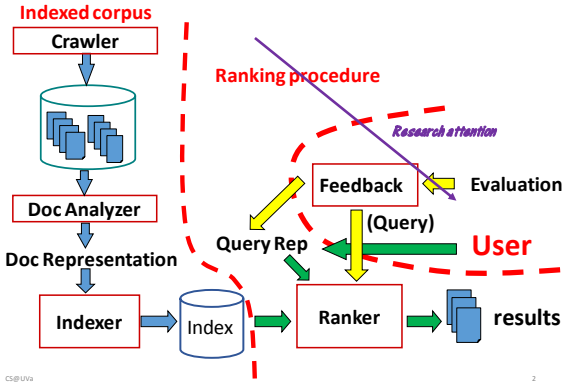
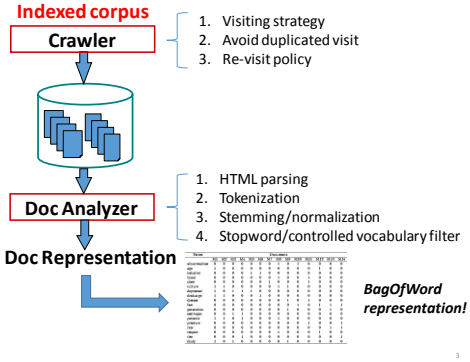
UIT
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

TRUY VẤN THÔNG TIN ĐA PHƯƠNG TIỆN
INFORMATION RETRIEVAL

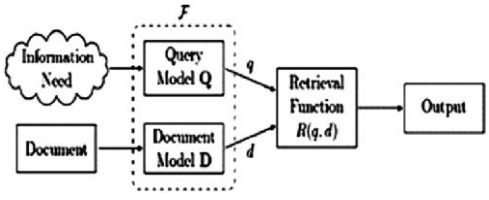
Retrieval Model





Mô hình truy vấn - Retrieval model

Algorithms that find the ***most relevant*** documents for the given information need



Một số mô hình trong IR

- Boolean model
  - simple model based on set theory
  - queries as Boolean expressions
  - adopted by many commercial systems
- Vector space model
  - queries and documents as vectors in an *M*-dimensional space
  - *M* is the number of terms
  - find documents most similar to the query in the *M*-dimensional space
- Probabilistic model
  - a probabilistic approach
  - assume an *ideal* answer set for each query
  - iteratively refine the properties of the ideal answer set

5

1. Mô hình Boolean model

- Mỗi văn bản được biểu diễn bằng một tập từ khóa
- Câu truy vấn là biểu thức Boolean của các từ khóa, kết nối với nhau bằng các phép AND, OR và NOT
- Đầu ra: Văn bản phù hợp hoặc không phù hợp
  - Không đối sánh một phần
  - Không phân hạng

6

1. Mô hình Boolean model

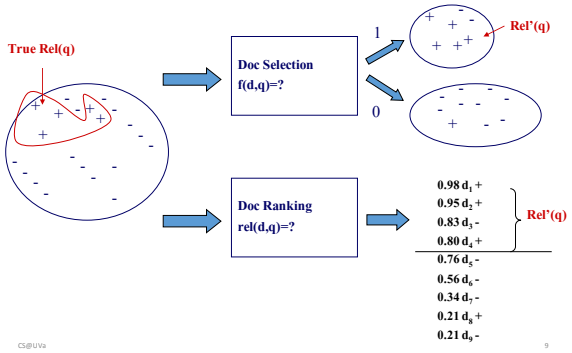
- Boolean query
  - E.g., "obama" AND "healthcare" NOT "news"
- Procedures
  - Lookup query term in the dictionary
  - Retrieve the posting lists
  - Operation
    - AND: intersect the posting lists
    - OR: union the posting list
    - NOT: diff the posting list

CS@UVa 7

1. Mô hình Boolean model

- Ưu điểm:
  - Dễ hiểu khi câu truy vấn đơn giản
- Cứng nhắc: AND nghĩa là tất cả, OR nghĩa là bất kỳ
- Khó diễn tả nhu cầu phức tạp của người dùng
- Khó kiểm soát số văn bản trả về:
  - Tất cả các văn bản khớp với câu truy vấn phải được trả về
- Khó phân hạng đầu ra:
  - Tất cả các văn bản trả về đều thỏa mãn câu truy vấn như nhau

Selection vs. Ranking



Ranking is often preferred

- Relevance is a matter of degree
  - Easier for users to find appropriate queries
- A user can stop browsing anywhere, so the boundary is controlled by the user
  - Users prefer coverage would view more items
  - Users prefer precision would view only a few

Vector Space Model

Relevance = Similarity

- Assumptions
  - Query and documents are represented in the same form
    - A query can be regarded as a "document"
  - Relevance(d,q) $\propto$ similarity(d,q)
- $R(q) = \{d \in C \mid rel(d,q) > \theta\}$, $rel(q,d) = \Delta(Rep(q), Rep(d))$
- Key issues
  - How to represent query/document?
  - How to define the similarity measure $\Delta(x,y)$?

Vector space model

- Represent both doc and query by concept vectors
  - Each concept defines one dimension
  - *K* concepts define a high-dimensional space
  - Element of vector corresponds to concept weight
    - E.g., d=(x_1,…,x_k), x_i is "importance" of concept i
- Measure relevance
  - Distance between the query vector and document vector in this concept space

Vector space model

- Vocabulary V={w_1, w_2, …, w_N} of language
- Query q = t_1,…,t_m, where $t_i \in V$
- Document d_i = t_{i1},…,t_{in}, where $t_{ij} \in V$
- Collection C= {d_1, …, d_k}
- Rel(q,d): relevance of doc d to query q
- Rep(d): representation of document d
- Rep(q): representation of query q

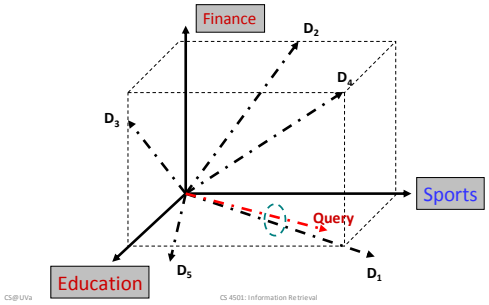
Vector space model

- Sau bước tiền xử lý văn bản, ta thu được **bộ từ vựng** gồm *t* từ khóa
- Không gian véctơ gồm *t* chiều, mỗi chiều ứng với một từ khóa
- Mỗi từ khóa *i* trong một văn bản hay câu truy vấn *j* có trọng số w_{ij} (là số thực)
- Mỗi văn bản và câu truy vấn được biểu diễn bằng một véctơ *t* chiều:

$d_j = (w_{1j}, w_{2j}, …, w_{tj})$

VS Model: an illustration

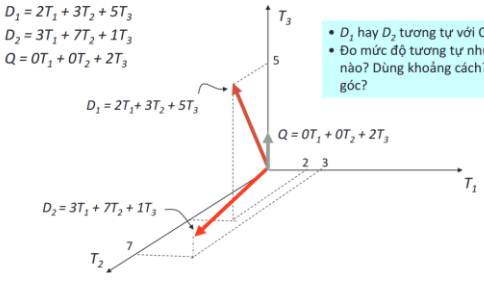
- Which document is closer to the query?



Vector space model

Ví dụ:

$D_1 = 2T_1 + 3T_2 + 5T_3$
$D_2 = 3T_1 + 7T_2 + 1T_3$
$Q = 0T_1 + 0T_2 + 2T_3$



- D_1 hay D_2 tương tự với Q hơn?
- Đo mức độ tương tự như thế nào? Dùng khoảng cách? Dùng góc?

17

Vector space model

- Biểu diễn một tập *n* văn bản trong mô hình không gian véctơ bằng một **ma trận từ khóa – văn bản**
- Mỗi phần tử trong ma trận là **trọng số** của một từ khóa trong văn bản: giá trị 0 nghĩa là từ khóa đó không tồn tại trong văn bản

	T_1	T_2	...	T_t
D_1	w_{11}	w_{21}	...	w_{t1}
D_2	w_{12}	w_{22}	...	w_{t2}
:	:	:		:
:	:	:		:
D_n	w_{1n}	w_{2n}	...	w_{tn}

18

What the VS model doesn't say

- How to define/select the "basic concept"
  - Concepts are assumed to be **orthogonal**
- How to assign weights
  - Weight in query indicates importance of the concept
  - Weight in doc indicates how well the concept characterizes the doc
- How to define the similarity/distance measure

CS@UVa 19

What is a good "basic concept"?

- Orthogonal
  - Linearly independent basis vectors
    - "Non-overlapping" in meaning
    - No ambiguity
- Weights can be assigned automatically and accurately
- Existing solutions
  - Terms or N-grams, i.e., bag-of-words
  - Topics, i.e., topic model

CS@UVa 20

How to assign weights?

- Important!
- Why?
  - Query side: not all terms are equally important
  - Doc side: some terms carry more information about the content
- How?
  - Two basic heuristics
    - TF (Term Frequency) = Within-doc-frequency
    - IDF (Inverse Document Frequency)
    - TF -IDF

TF weighting

- Idea: a term is more important if it occurs more frequently in a document
- TF Formulas
  - Let $f(t,d)$ be the frequency count of term t in doc d
  - Raw TF: $tf(t,d) = f(t,d)$
- Từ khóa xuất hiện thường xuyên hơn trong một văn bản sẽ quan trọng hơn vì nó chỉ báo nhiều hơn về chủ đề của văn bản:

  f_{ij} = tần số của từ khóa *i* trong văn bản *j*

TF normalization

- Two views of document length
  - A doc is long because it is verbose
  - A doc is long because it has more content
- Raw TF is inaccurate
  - Document length variation
  - "Repeated occurrences" are less informative than the "first occurrence"
  - Relevance does not increase proportionally with number of term occurrence
- Generally penalize long doc, but avoid over-penalizing
  - Pivoted length normalization

TF normalization - scaled frequency

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

- tf(t, d): tần suất xuất hiện của từ t trong văn bản d
- f(t, d): Số lần xuất hiện của từ t trong văn bản d
- max({f(w, d) : w ∈ d}): Số lần xuất hiện của từ có số lần xuất hiện nhiều nhất trong văn bản d

TF normalization - scaled length

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\text{\# of words in a document}}$$

- tf(t, d): tần suất xuất hiện của từ t trong văn bản d
- f(t, d): Số lần xuất hiện của từ t trong văn bản d

Document frequency

- Idea: a term is more discriminative if it occurs only in fewer documents

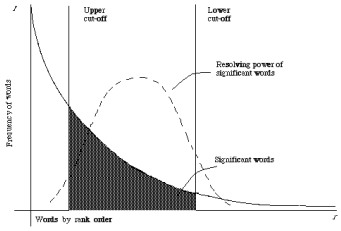


Figure 2.1. A plot of the hyperbolic curve relating f, the frequency of occurrence and r, the rank order (Adapted from Schultz [1968, page 120])

Document frequency

df_i = tần số văn bản (document frequency) của từ khóa *i*
= số văn bản chứa từ khóa *i*

IDF weighting

- Solution
  - Assign higher weights to the rare terms
  - Formula
    - $IDF(t) = 1 + \log(\frac{N}{df(t)})$
      - Non-linear scaling
      - Total number of docs in collection
      - Number of docs containing term t
  - A corpus-specific property
    - Independent of a single document

Example

Document 1	Người lên ngựa kẻ chia bào. Rừng phong thu đã nhốm màu quan san
Document 2	Ô hay buồn vương cây ngô đồng. Vàng rơi vàng rơi thu mênh mông
Document 3	Một chiều về bên bến sông thu. Nghe tin em cưới á cái đù.

29

Example

TF của Document 1

Document 1	người	lên	ngựa	kẻ	chia	bào	rừng
Term Frequency	1	1	1	1	1	1	1
	phong	thu	đã	nhốm	màu	quan	san
	1	1	1	1	1	1	1

30

Example

TF Document 2

Document 2	ô	hay	buồn	vương	cây	ngô
Term Frequency	1	1	1	1	1	1
	đồng	vàng	rơi	thu	mênh	mông
	1	2	2	1	1	1

31

Example

TF của Document 3

Document 3	một	chiều	về	bên	bến	sông	thu
Term Frequency	1	1	1	1	1	1	1
	nghe	tin	em	cưới	á	cái	đù
	1	1	1	1	1	1	1

32

Example

TF của Document 1

Document 1	người	lên	ngựa	kẻ	chia	bào	rừng
Term Frequency	1	1	1	1	1	1	1
	phong	thu	đã	nhốm	màu	quan	san
	1	1	1	1	1	1	1

Normalized TF cho Document 1:

Document 1	người	lên	ngựa	kẻ	chia	bào	rừng
Term Frequency	0.07	0.07	0.07	0.07	0.07	0.07	0.07
	phong	thu	đã	nhốm	màu	quan	san
	0.07	0.07	0.07	0.07	0.07	0.07	0.07

33

Example

TF Document 2

Document 2	ô	hay	buồn	vương	cây	ngô
Term Frequency	1	1	1	1	1	1
	đồng	vàng	rơi	thu	mênh	mông
	1	2	2	1	1	1

Normalized TF cho Document 2:

Document 2	ô	hay	buồn	vương	cây	ngô
Term Frequency	0.07	0.07	0.07	0.07	0.07	0.07
	đồng	vàng	rơi	thu	mênh	mông
	0.07	0.14	0.14	0.07	0.07	0.07

34

Example

TF của Document 3

Document 3	một	chiều	về	bên	bến	sông	thu
Term Frequency	1	1	1	1	1	1	1
	nghe	tin	em	cười	á	cái	đú
	1	1	1	1	1	1	1

Normalized TF cho Document 3:

Document 3	một	chiều	về	bên	bến	sông	thu
Term Frequency	0.07	0.07	0.07	0.07	0.07	0.07	0.07
	nghe	tin	em	cười	á	cái	đú
	0.07	0.07	0.07	0.07	0.07	0.07	0.07

35

Example

```
IDF(chiều) = 1 + ln(Tổng số văn bản trong data set/Số văn bản chứa từ chiều)

Data set của chúng ta có 3 văn bản : Document 1, Document 2 và Document 3.

Từ chiều xuất hiện trong Document 3.

IDF(chiều) = 1 + ln(3/1) = 1 + 1.0986 = 2.0986.
```

36

Example

Từ	IDF
người	2.0986
lên	2.0986
ngựa	2.0986
kẻ	2.0986
chia	2.0986
bào	2.0986
rừng	2.0986
phong	2.0986
thu	1
đã	2.0986
nhốm	2.0986
màu	2.0986
quan	2.0986
san	2.0986
ô	2.0986
hay	2.0986
buồn	2.0986
vương	2.0986
cây	2.0986
ngô	2.0986
đồng	2.0986
vàng	2.0986
rơi	2.0986
mênh	2.0986
mông	2.0986
một	2.0986
chiều	2.0986
về	2.0986
bên	2.0986
bến	2.0986
sông	2.0986
nghe	2.0986
tin	2.0986
em	2.0986
cưới	2.0986
á	2.0986
cái	2.0986
đù	2.0986

37

TF-IDF weighting

- Combining TF and IDF
  - Common in doc → high tf → high weight
  - Rare in collection→ high idf→ high weight
  - $w(t,d) = TF(t,d) \times IDF(t)$
- Most well-known document representation schema in IR! (G Salton et al. 1983)

"Salton was perhaps the leading computer scientist working in the field of information retrieval during his time." - wikipedia

Gerard Salton Award – highest achievement award in IR

CS@UVa CS 4501: Information Retrieval 38

TF-IDF

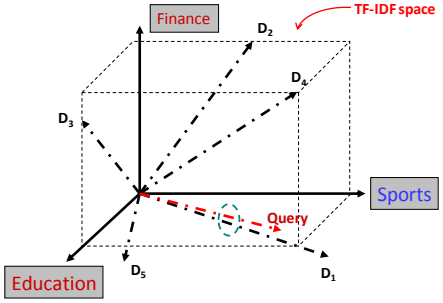
- Kết hợp các trọng số *tf* và *idf*:

$$w_{ij} = tf_{ij}\, idf_i = tf_{ij} \log_2 (N / df_i)$$

- Ý nghĩa: Từ khóa xuất hiện thường xuyên hơn trong một văn bản nhưng hiếm thấy hơn trong các văn bản còn lại sẽ quan trọng hơn (có trọng số cao hơn)
- Thực nghiệm cho thấy trọng số từ TF-IDF thường làm việc tốt

39

How to define a good similarity measure?



CS@UVa 40

Similarity measure

- Độ đo tương tự là một hàm tính mức độ tương tự giữa hai véctơ
- Dùng độ đo tương tự giữa câu truy vấn và mỗi văn bản:
  - Phân hạng các văn bản trả về theo mức độ tương tự
  - Có thể đặt ra ngưỡng để kiểm soát số văn bản trả về

41

Similarity measure

Độ đo tương tự tích trong

- Có thể tính độ tương tự giữa véctơ văn bản $\boldsymbol{d}_j$ và véctơ truy vấn $\boldsymbol{q}$ bằng tích trong của hai véctơ đó:

$$\text{sim}(\boldsymbol{d}_j, \boldsymbol{q}) = \boldsymbol{d}_j \bullet \boldsymbol{q} = \sum_{i=1}^{t} w_{ij} w_{iq}$$

  trong đó w_{ij} là trọng số của từ i trong văn bản j và w_{iq} là trọng số của từ i trong câu truy vấn q
- Đối với các véctơ nhị phân, tích trong bằng số từ khóa truy vấn xuất hiện trong văn bản (kích thước phần giao)
- Đối với các véctơ có trọng số, tích trong là tổng tích của các trọng số của các từ khóa xuất hiện đồng thời trong cả văn bản và câu truy vấn

Similarity measure

Các tính chất của tích trong

- Tích trong không bị chặn
- Thiên vị những văn bản dài và chứa một số lượng lớn các từ khóa riêng biệt
- Đo bao nhiêu từ khóa khớp được nhưng không đo bao nhiêu từ khóa không khớp nhau giữa văn bản và câu truy vấn

43

Tích trong – ví dụ

Nhị phân:

	retrieval	database	architecture	computer	text	management	information
D =	1,	1,	1,	0,	1,	1,	0
Q =	1,	0,	1,	0,	0,	1,	1

sim(D, Q) = 3

Kích thước véctơ = Kích thước bộ từ vựng = 7

Giá trị 0 nghĩa là từ khóa tương ứng vắng mặt trong văn bản hoặc câu truy vấn

Có trọng số:

$D_1 = 2T_1 + 3T_2 + 5T_3 \quad D_2 = 3T_1 + 7T_2 + 1T_3$

$Q = 0T_1 + 0T_2 + 2T_3$

$\text{sim}(D_1, Q) = 2*0 + 3*0 + 5*2 = 10$

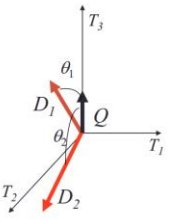
$\text{sim}(D_2, Q) = 3*0 + 7*0 + 1*2 = 2$

44

Độ đo tương tự côsin

- Đo côsin của góc giữa hai véctơ
- Tích trong được chuẩn hóa bằng chiều dài của các véctơ

$$\text{CosSim}(\vec{d}_j, \vec{q}) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^{t}(w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}}$$



$D_1 = 2T_1 + 3T_2 + 5T_3$ $\text{CosSim}(D_1, Q) = 10 / \sqrt{(4+9+25)(0+0+4)} = 0{,}81$
$D_2 = 3T_1 + 7T_2 + 1T_3$ $\text{CosSim}(D_2, Q) = 2 / \sqrt{(9+49+1)(0+0+4)} = 0{,}13$
$Q = 0T_1 + 0T_2 + 2T_3$

Nhận xét: D_1 phù hợp hơn D_2 6 lần khi dùng côsin, nhưng chỉ 5 lần khi dùng tích trong (xem slide trước)

45

Mô hình không gian vector

- Biến đổi tất cả các văn bản $\boldsymbol{d}_j$ trong tập văn bản D thành các véctơ có trọng số TF-IDF dùng bộ từ vựng V
- Biến đổi câu truy vấn $\boldsymbol{q}$ thành véctơ có trọng số TF-IDF
- Đối với mỗi văn bản $\boldsymbol{d}_j$ trong D:
  - Tính điểm số $s_j = \text{CosSim}(\boldsymbol{d}_j, \boldsymbol{q})$
- Sắp xếp các văn bản theo thứ tự điểm số giảm dần
- Trình diễn các tài liệu được xếp hạng cao nhất cho người dùng

46

Mô hình Vector – ưu điểm

- Cách tiếp cận đơn giản, dựa trên toán học
- Xem xét tần số xuất hiện của các từ khóa vừa cục bộ (*tf*) vừa toàn cục (*idf*)
- Cho phép đối sánh một phần và phân hạng kết quả
- Thường làm việc khá tốt trong thực tế

[No Title]

47

Tài liệu tham khảo

Slide được tham khảo từ:

- http://www.cs.virginia.edu/~hw5x/Course/IR2015/_site/lectures/
- https://nlp.stanford.edu/IR-book/newslides.html
- https://course.ccs.neu.edu/cs6200s14/slides.html