Truy hồi Boole có phân hạng và mô hình không gian vector

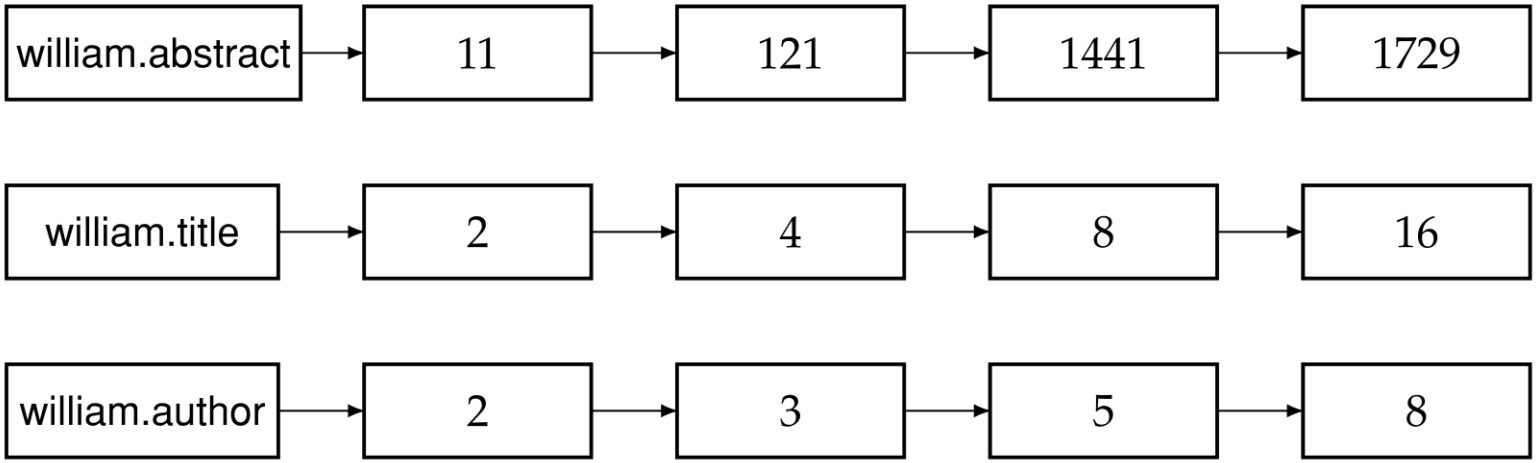
Nguyễn Mạnh Hiển
hiennm@tlu.edu.vn

Chỉ mục vùng (zone index)

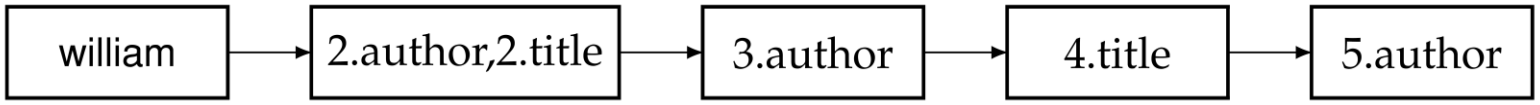
- Mỗi văn bản có thể bao gồm một số vùng (zone):
  - Tiêu đề (title)
  - Tóm tắt (abstract)
  - Tác giả (author)
  - Thân (body)
- Câu truy vấn có thể dưới dạng: Tìm các văn bản có từ merchant trong tiêu đề và từ william trong danh sách tác giả và cụm từ gentle rain trong phần thân.

Chỉ mục vùng (tiếp)

Mã hóa vùng vào trong từ điển:



Mã hóa vùng vào trong các thông báo (posting):



Chấm điểm vùng có trọng số

- Cho câu truy vấn Boole q và văn bản d.
- Gán cho cặp (q, d) một điểm số trong khoảng [0; 1]:

$$\sum_{i=1}^{l} g_i s_i$$

  - g_i là trọng số của vùng i: $\sum_{i=1}^{l} g_i = 1$.
  - s_i là điểm số Boole của vùng i, bằng 1 hoặc 0 tùy theo các từ truy vấn xuất hiện hoặc vắng mặt trong vùng i.
- Chấm điểm vùng có trọng số còn được gọi là **truy hồi Boole có phân hạng** (ranked Boole retrieval).

Chấm điểm vùng có trọng số: Ví dụ

- Mỗi văn bản có ba vùng:
  - author: $g_1 = 0,2$
  - title: $g_2 = 0,3$
  - body: $g_3 = 0,5$
- Xét câu truy vấn shakespeare.
- Nếu từ shakespeare xuất hiện trong một vùng thì điểm số Boole cho vùng đó là 1, ngược lại là 0.
- Giả sử một văn bản có từ shakespeare trong các vùng title và body, nhưng không có từ đó trong vùng author.

$\rightarrow$ Điểm số của văn bản này = 0*0,2 + 1*0,3 + 1*0,5 = 0,8.

Thuật toán tính điểm vùng có trọng số

- Xét câu truy vấn gồm hai từ q_1 và q_2.
- Nếu cả hai từ q_1 và q_2 đều xuất hiện trong một vùng thì điểm số Boole của vùng đó là 1, ngược lại là 0.
- Thuật toán chi tiết… (xem slide sau)

Thuật toán tính điểm vùng có trọng số

ZoneScore(q_1, q_2)

```
1  float scores[N] = [0]
2  constant g[ℓ]
3  p1 ← postings(q1)
4  p2 ← postings(q2)
5  // scores[] is an array with a score entry for each document, initialized to zero.
6  //p1 and p2 are initialized to point to the beginning of their respective postings.
7  //Assume g[] is initialized to the respective zone weights.
8  while p1 ≠ NIL and p2 ≠ NIL
9  do if docID(p1) = docID(p2)
10       then scores[docID(p1)] ← WeightedZone(p1, p2, g)
11            p1 ← next(p1)
12            p2 ← next(p2)
13       else if docID(p1) < docID(p2)
14            then p1 ← next(p1)
15            else p2 ← next(p2)
16 return scores
```

Học các trọng số (learning weights)

- Tính các trọng số g_i cho các vùng như thế nào?
  - Cách 1: Chuyên gia chỉ định.
  - Cách 2: Người dùng chỉ định.
  - Cách 3: **Học các trọng số** từ tập huấn luyện.
- **Tập huấn luyện** (training set) gồm các **mẫu huấn luyện** (training example). Mẫu huấn luyện là một bộ ba:
  1. Câu truy vấn q.
  2. Văn bản d.
  3. Đánh giá d có **phù hợp** (relevant) với q hay không: phù hợp / không phù hợp.

Minh họa tập huấn luyện

- Xét các văn bản có hai vùng title và body.
- $s_T(d, q)$ là điểm số Boole (1/0) cho vùng title.
- $s_B(d, q)$ là điểm số Boole (1/0) cho vùng body.
- Mỗi mẫu huấn luyện là một bộ ba $\Phi = (d, q, r(d, q))$, trong đó:
  - $r(d, q) = 1$ nếu phù hợp (relevant);
  - $r(d, q) = 0$ nếu không phù hợp (nonrelevant).

Example	DocID	Query	s_T	s_B	Judgment
Φ_1	37	linux	1	1	Relevant
Φ_2	37	penguin	0	1	Non-relevant
Φ_3	238	system	0	1	Relevant
Φ_4	238	penguin	0	0	Non-relevant
Φ_5	1741	kernel	1	1	Relevant
Φ_6	2094	driver	0	1	Relevant
Φ_7	3191	driver	1	0	Non-relevant

Tính điểm số và sai số

- Với mỗi mẫu huấn luyện Φ_j, tính điểm số vùng có trọng số:

$$score(d_j, q_j) = g \cdot s_T(d_j, q_j) + (1 - g) \cdot s_B(d_j, q_j)$$

- Sai số (bình phương):

$$\varepsilon(g, \Phi_j) = (r(d_j, q_j) - score(d_j, q_j))^2$$

- Tổng sai số:

$$\sum_j \varepsilon(g, \Phi_j)$$

$\rightarrow$ Cần chọn giá trị của g để cực tiểu hóa tổng sai số.

Tìm trọng số g tối ưu

s_T	s_B	Score
0	0	0
0	1	$1-g$
1	0	g
1	1	1

- Gọi n_{01r} (n_{01n}) là số mẫu huấn luyện có:
  - $s_T = 0$ và $s_B = 1$;
  - đánh giá là phù hợp (không phù hợp).
- Sai số đóng góp bởi những mẫu có $s_T = 0$ và $s_B = 1$:

$$[1 - (1 - g)]^2 n_{01r} + [0 - (1 - g)]^2 n_{01n}$$

- Tương tự, ta viết được sai số cho ba trường hợp còn lại, rồi cộng tất cả lại để được tổng sai số:

$$(n_{01r} + n_{10n})g^2 + (n_{10r} + n_{01n})(1 - g)^2 + n_{00r} + n_{11n}$$

- Lấy đạo hàm tổng sai số rồi cho bằng 0, ta được giá trị tối ưu của g:

$$\frac{n_{10r} + n_{01n}}{n_{10r} + n_{10n} + n_{01r} + n_{01n}}$$

Bài tập truy hồi Boole có phân hạng

1. Khi chấm điểm vùng có trọng số, có nhất thiết phải dùng cùng một cách tính điểm số Boole cho các vùng khác nhau hay không?

2. Xét tập văn bản gồm ba vùng author, title và body với các trọng số tương ứng là $g_1 = 0,2$, $g_2 = 0,31$ và $g_3 = 0,49$. Xét câu truy vấn shakespeare; nếu từ shakespeare xuất hiện trong vùng nào thì điểm số Boole của vùng đó bằng 1, ngược lại bằng 0. Hãy liệt kê tất cả những điểm số khác nhau mà một văn bản có thể nhận được.

Bài tập truy hồi Boole có phân hạng

3. Áp dụng công thức ở slide 11 để tìm giá trị tối ưu của g cho tập huấn luyện ở slide 9.

4. Với g tính được ở bài tập 3, hãy tính điểm vùng có trọng số cho mỗi cặp <văn bản, câu truy vấn> trong bảng ở slide 9. Nhận xét về mối quan hệ của các điểm số tính được với các đánh giá về sự phù hợp đã biết.

Định trọng số từ (term weighting)

- Truy hồi Boole chỉ xem xét sự có mặt hoặc vắng mặt của từ trong văn bản.
- Định trọng số từ là cách gán một giá trị (số thực không âm) cho mỗi từ dựa trên số lần xuất hiện của từ đó nhằm phản ánh tầm quan trọng của từ đó.
- Ba cách định trọng số từ:
  - Tần số từ tf (term frequency)
  - Tần số văn bản nghịch đảo idf (inverse document frequency)
  - tf-idf (kết hợp của hai cách trên)

Tần số từ

- Ký hiệu là $tf_{t,d}$: Số lần từ t xuất hiện trong văn bản d.
- Thứ tự xuất hiện của các từ bị bỏ qua $\rightarrow$ **Mô hình túi từ** (bag of words).
- Hai văn bản sau đây là như nhau:

  Doc1: “Mary is quicker than John”

  Doc2: “John is quicker than Mary”

Tần số văn bản

- Xét tập văn bản về ngành công nghiệp ô tô:
  - Từ auto xuất hiện trong hầu như mọi văn bản nên không có tác dụng phân biệt nội dung của các văn bản khác nhau.
- Tần số văn bản df_t (document frequency) là số văn bản chứa từ t đang xét.
  - Từ xuất hiện trong càng ít văn bản, tức là df nhỏ, thì sức phân biệt nội dung của từ đó càng lớn.

Tần số văn bản nghịch đảo

- Tần số văn bản nghịch đảo (inverse document frequency):

$$idf_t = \log \frac{N}{df_t}$$

  - N là tổng số văn bản.
  - Cơ số của hàm lôgarít không quan trọng.
- Tầm quan trọng của từ tỉ lệ với tần số văn bản nghịch đảo.

term	df_t	idf_t
car	18,165	1.65
auto	6723	2.08
insurance	19,241	1.62
best	25,235	1.5

Tập bài báo tin tức Reuters gồm 806.791 văn bản; cơ số của hàm lôgarít là 10.

Trọng số từ tf-idf

- Sự kết hợp của tần số từ tf và tần số văn bản nghịch đảo idf:

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t$$

  - Từ xuất hiện thường xuyên trong một văn bản nhưng xuất hiện trong không nhiều các văn bản khác thì quan trọng đối với văn bản đang xét.
- Xét câu truy vấn q, ta có thể định nghĩa điểm số của văn bản d bằng tổng các giá trị tf-idf của các từ trong q có xuất hiện trong d:

$$\text{score}(q,d) = \sum_{t \in q} \text{tf-idf}(t,d)$$

Bài tập định trọng số từ

1. Tại sao idf của một từ luôn hữu hạn?

2. Nếu một từ xuất hiện trong mọi văn bản thì idf của từ đó bằng bao nhiêu? So sánh điều này với việc sử dụng danh sách từ dừng (stop word).

Bài tập định trọng số từ

3. Tính trọng số tf-idf cho các từ car, auto, insurance và best trong mỗi văn bản Doc1, Doc2 và Doc3.

	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

term	df_t	idf_t
car	18,165	1.65
auto	6723	2.08
insurance	19,241	1.62
best	25,235	1.5

Bài tập định trọng số từ

4. Trọng số tf-idf của một từ trong một văn bản có thể vượt quá 1 hay không?

5. Cơ số của hàm lôgarít trong công thức tính idf ảnh hưởng như thế nào đến việc tính điểm số truy vấn cho các văn bản theo công thức bên dưới? Cơ số của hàm lôgarít ảnh hưởng như thế nào đến điểm số tương đối của hai văn bản cho cùng một câu truy vấn?

$$\text{score}(q, d) = \sum_{t \in q} \text{tf-idf}(t, d)$$

Mô hình không gian vector

- Các văn bản trở thành các vector trong không gian vector với mỗi trục ứng với mỗi từ trong từ điển.
- Câu truy vấn cũng được xem là một vector trong không gian vector đó.
- Tính toán trên các văn bản và câu truy vấn quy về tính toán trên các vector.

Độ tương tự giữa hai văn bản

- Gọi $\vec{V}(d)$ là vector biểu diễn văn bản d:
  - Mỗi thành phần của vector là trọng số (ví dụ, tf-idf) của một từ trong từ điển.
- Tính độ tương tự giữa hai văn bản trong không gian vector như thế nào?
  - Một cách là tính chiều dài của vector hiệu (độ lớn của sai khác) $\rightarrow$ Phân phối từ trong hai văn bản có thể giống nhau nhưng chiều dài khác hẳn nhau!
  - Cách tốt hơn là tính độ tương tự côsin.

Độ tương tự côsin

- Độ tương tự côsin (của góc giữa hai vector):

$$\text{sim}(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{\left|\vec{V}(d_1)\right|\left|\vec{V}(d_2)\right|}$$

- Chú ý rằng $\vec{v}(d_1) = \vec{V}(d_1)/\left|\vec{V}(d_1)\right|$ và $\vec{v}(d_2) = \vec{V}(d_2)/\left|\vec{V}(d_2)\right|$ là các vector đơn vị của các văn bản d_1 và d_2.

  $\rightarrow$ Đó là sự **chuẩn hóa vector** bằng chiều dài của nó.

- Có thể viết lại công thức tính độ tương tự côsin như sau:

$$\text{sim}(d_1, d_2) = \vec{v}(d_1) \cdot \vec{v}(d_2)$$

  $\rightarrow$ Đó là tích vô hướng của hai vector đơn vị.

Chuẩn hóa thành các vector đơn vị

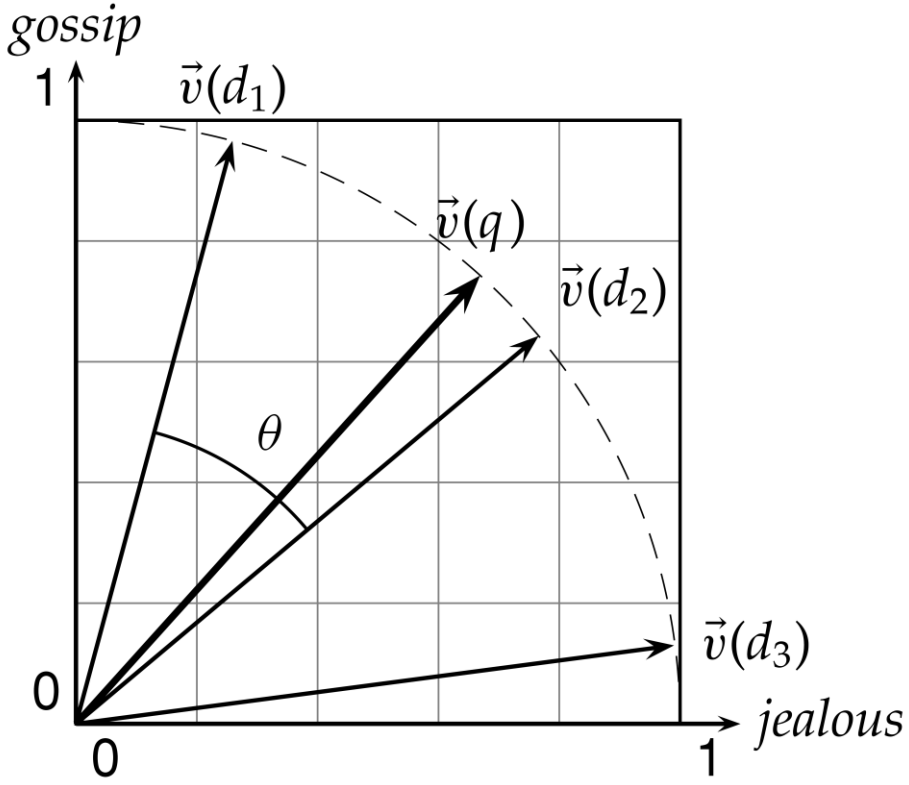
Hãy chuẩn hóa các véctơ sau!

	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

(Các thành phần của các vector là các tần số từ tf)

Minh họa độ tương tự côsin

- Thực chất là côsin của góc giữa hai vector.
- Nêu một kịch bản ứng dụng việc tính độ tương tự giữa hai văn bản?



Ví dụ

- Xét ba tiểu thuyết:
  - Sense and Sensibility (SaS)
  - Pride and Prejudice (PaP)
  - Wuthering Heights (WH)
- Tính được:

$\text{sim}(\vec{v}(\text{SaS}), \vec{v}(\text{PaP})) = 0.999$

$\text{sim}(\vec{v}(\text{SaS}), \vec{v}(\text{WH})) = 0.888$

$\rightarrow$ Hai tiểu thuyết SaS và PaP giống nhau hơn.

term	SaS	PaP	WH
affection	115	58	20
jealous	10	7	11
gossip	2	0	6

Chuẩn hóa thành véctơ đơn vị

term	SaS	PaP	WH
affection	0.996	0.993	0.847
jealous	0.087	0.120	0.466
gossip	0.017	0	0.254

Câu truy vấn cũng là vector

- Có thể xem câu truy vấn là một văn bản rất ngắn → Câu truy vấn cũng biểu diễn được thành một vector.
- Tiếp tục ví dụ ở slide trước:
  - Câu truy vấn q = jealous gossip biến thành vector đơn vị $\vec{v}(q)$ = (0, 0.707, 0.707).
  - Tính độ tương tự côsin $\vec{v}(d) \cdot \vec{v}(q)$của mỗi tiểu thuyết d với câu truy vấn q, ta được:
    - Điểm số cho WH là 0.509.
    - Điểm số cho PaP là 0.085.
    - Điểm số cho SaS là 0.074.

Ví dụ thêm

- Tập văn bản có N = 1,000,000 văn bản.
- Câu truy vấn best car insurance.

term	query				document			product
	tf	df	idf	$w_{t,q}$	tf	wf	$w_{t,d}$	
auto	0	5000	2.3	0	1	1	0.41	0
best	1	50000	1.3	1.3	0	0	0	0
car	1	10000	2.0	2.0	1	1	0.41	0.82
insurance	1	1000	3.0	3.0	2	2	0.82	2.46

Thuật toán tính điểm số côsin

CosineScore(q)

1 float $Scores[N] = 0$
2 Initialize $Length[N]$
3 **for each** query term t
4 **do** calculate $w_{t,q}$ and fetch postings list for t
5 **for each** pair(d, $tf_{t,d}$) in postings list
6 **do** $Scores[d] \mathrel{+}= wf_{t,d} \times w_{t,q}$
7 Read the array $Length[d]$
8 **for each** d
9 **do** $Scores[d] = Scores[d]/Length[d]$
10 **return** Top K components of $Scores[]$

Bài tập mô hình không gian vector

1. Giả sử ta phải tách gốc các từ jealous và jealousy để quy chúng về một gốc từ chung trước khi tạo lập không gian vector. Hỏi các giá trị tf và idf của các từ đó cần được thay đổi như thế nào?

2. Xét các trọng số tf-idf tính được trong bài tập ở slide 20. Hãy chuẩn hóa các vector văn bản theo chiều dài của chúng.

3. Kiểm chứng tổng bình phương của các thành phần của mỗi vector tính được trong bài tập 2 bằng 1 (với một sai số làm tròn nào đó).

Bài tập mô hình không gian vector

4. Với các trọng số từ tính được trong bài tập 2, hãy phân hạng các văn bản theo điểm số côsin cho câu truy vấn car insurance cho mỗi cách định trọng số từ trong câu truy vấn sau đây:

(a) Trọng số từ bằng 1 nếu từ có mặt trong câu truy vấn, ngược lại bằng 0.

(b) idf chuẩn hóa theo chiều dài.

Các biến thể của tf-idf

- Áp dụng hàm log lên tf
- Chuẩn hóa tf cực đại

Áp dụng hàm log lên tf

- Một từ xuất hiện 20 lần trong một văn bản không có nghĩa là tầm quan trọng của từ đó tăng lên 20 lần so với việc nó chỉ xuất hiện một lần.
- Áp dụng hàm log lên tf để giảm bớt độ tăng trưởng của tf:

$$\mathrm{wf}_{t,d} = \begin{cases} 1 + \log \mathrm{tf}_{t,d} & \text{nếu } \mathrm{tf}_{t,d} > 0 \\ 0 & \text{ngược lại} \end{cases}$$

- Trọng số tf-idf được thay bằng:

$$\text{wf-idf}_{t,d} = \mathrm{wf}_{t,d} \times \mathrm{idf}_t$$

Chuẩn hóa tf cực đại

- Gọi tf_{max}(d) là tần số từ lớn nhất trong văn bản d:

$$\text{tf}_{\max}(d) = \max_{\tau \in d} \text{tf}_{\tau,d}$$

- Chuẩn hóa tf cực đại (n trong ntf thay cho "normalized"):

$$\text{ntf}_{t,d} = a + (1-a)\frac{\text{tf}_{t,d}}{\text{tf}_{max}(d)}$$

  - Tham số a thường được đặt bằng 0.4, được dùng để hãm bớt số hạng thứ hai lại. Ví dụ, khi tf tăng 10 lần thì ntf cũng sẽ tăng 10 lần nếu không dùng tham số a.
  - Giúp khắc phục trường hợp bất thường: Một văn bản được lặp lại nhiều lần, tức là số lần xuất hiện của mỗi từ tăng lên nhưng không làm thay đổi nội dung văn bản.
  - Không tốt khi văn bản chứa một từ nhiều bất thường so với các từ còn lại.

Bài tập

1. Một độ đo tương tự khác giữa hai vector là khoảng cách Ơclít:

$$|\vec{x} - \vec{y}| = \sqrt{\sum_{i=1}^{M} (x_i - y_i)^2}$$

Cho một câu truy vấn q và các văn bản d_1, d_2, …, ta có thể phân hạng các văn bản d_i theo thứ tự khoảng cách Ơclít tăng dần.

Chứng minh rằng nếu q và các d_i được chuẩn hóa về vector đơn vị thì thứ hạng dựa trên khoảng cách Ơclít giống như thứ hạng dựa trên độ tương tư côsin.

Bài tập

2. Xét câu truy vấn digital cameras và văn bản digital cameras and video cameras. Giả sử tổng số văn bản N = 10,000,000; định trọng số tf với hàm $\log_{10}$ (các cột wf) cho cả câu truy vấn và văn bản; định trọng số idf chỉ cho câu truy vấn; chuẩn hóa vector theo chiều dài chỉ cho văn bản. Coi and là một từ dừng. Hãy điền vào các cột trống trong bảng. Tính điểm số tương tự côsin.

	query					document			
word	tf	wf	df	idf	q_i = wf-idf	tf	wf	d_i = normalized wf	$q_i \cdot d_i$
digital			10,000						
video			100,000						
cameras			50,000						

Bài tập

3. Xét câu truy vấn affection. Hãy tính điểm số tương tự côsin của các văn bản ở các slide 27-28 và cho thấy thứ tự của các điểm số ấy đảo ngược so với thứ tự điểm số cho câu truy vấn jealous gossip.

4. Xét trường hợp một từ truy vấn không có mặt trong M từ của chỉ mục ngược, và vì vậy vector truy vấn $\vec{V}(q)$ không nằm trong cùng không gian vector của tập văn bản. Hỏi ta nên xử lý tình huống này như thế nào?