

- Giới thiệu
- Kiến trúc hệ tìm kiếm thông tin
- Các Mô hình so khớp (Matching model)
- Đánh giá một hệ thống tìm kiếm thông tin
- Demo:
 - Xây dựng hệ tìm kiếm thông tin bằng **elasticsearch**
- Một số hướng tiếp cận mới

**Tìm kiếm thông tin
(Information Retrieval)**

Nguyễn Trường Sơn
ntson@fit.hcmus.edu.vn

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



Tìm kiếm thông tin là gì ?

- Mục tiêu của IR : Trả về các thông tin liên quan nhất đến nhu cầu thông tin của người dùng

Thông tin :

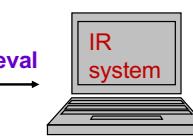
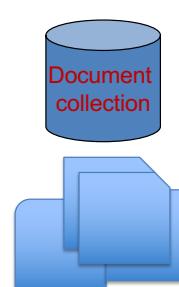
Một tài liệu trong các dạng khác nhau như:
sách, bài báo...

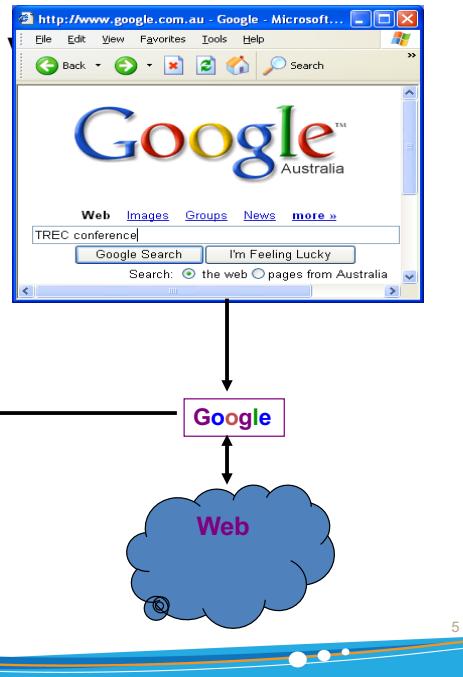
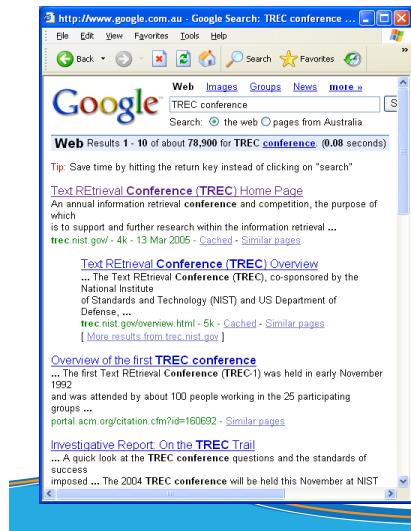
Một phần tử của một cấu trúc : một đoạn,
một câu

Q1 = “điện thoại iphone 12 pro max”
Q2 = “dân số việt nam là bao nhiêu”



Q1 = “điện thoại iphone 12 pro max” Info.
Q2 = “dân số việt nam là bao nhiêu” need





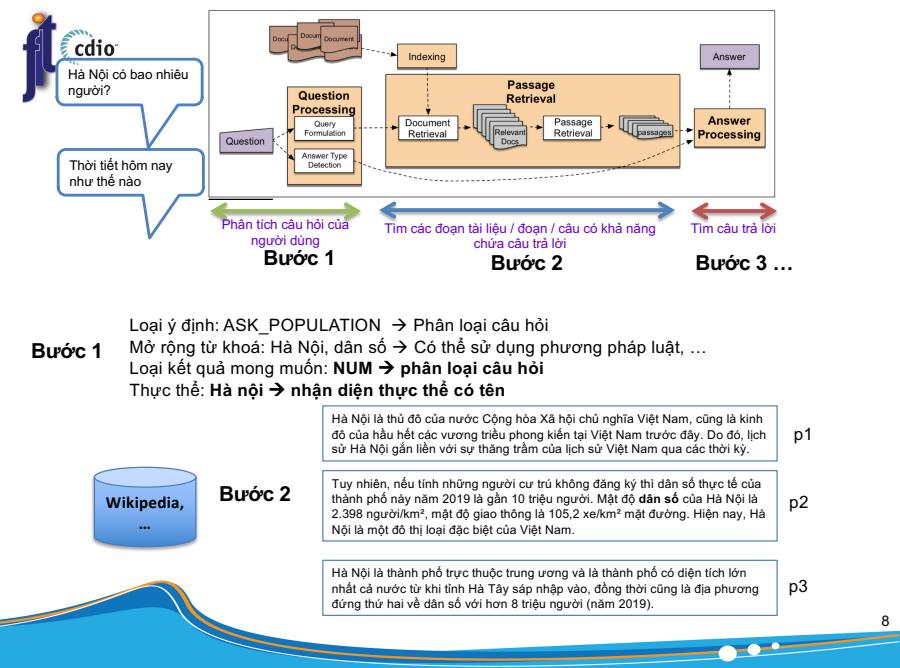
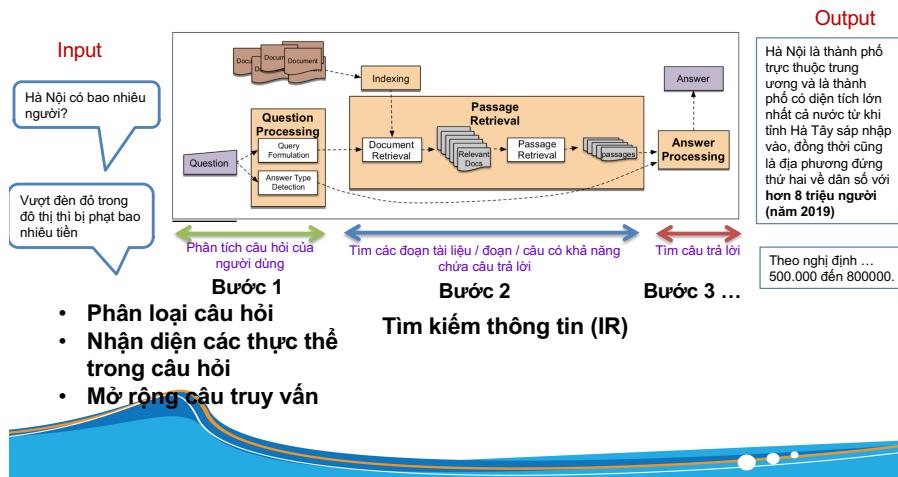
So sánh IR và Database

	Database	IR
Dữ liệu	Có cấu trúc	Phi cấu trúc
Trường	Có, ngữ nghĩa rõ ràng (ví dụ : HOTEN, PHAI)	Không có (chỉ là văn bản)
Câu truy vấn	Xác định trước theo một cấu trúc (Đại số quan hệ, SQL)	"ngôn ngữ tự nhiên"
So khớp	Chính xác (kết quả luôn luôn đúng)	Không chính xác (cần một độ đo)

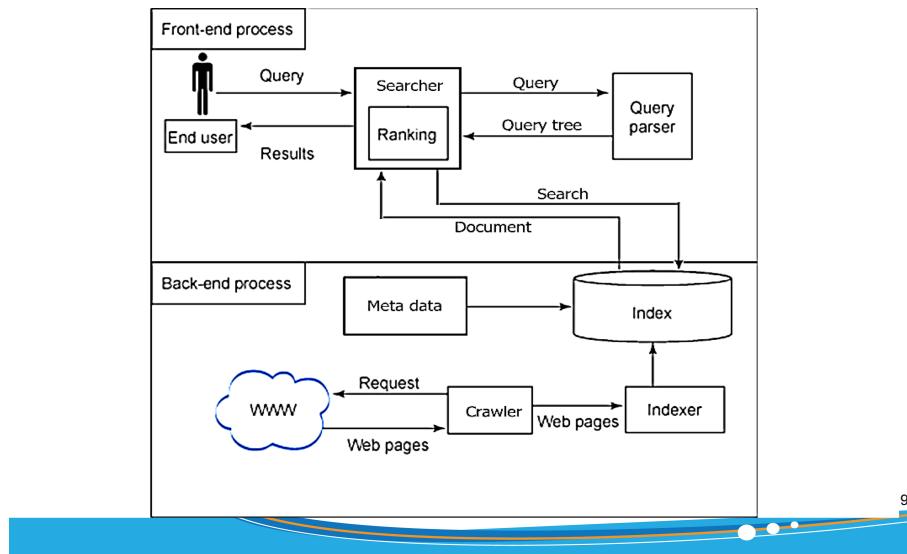


Ứng dụng của IR trong việc XD HTHD trên nguồn DL Văn bản

Khái niệm phổ biến: IR-based Question Answering



Web Search Process



Các vấn đề của IR

❑ Các ứng dụng đầu tiên trong lĩnh vực thư viện (1950)

ISBN: 0-201-12227-8
Author: Salton, Gerard
Title: Automatic text processing: the transformation, analysis, and retrieval of information by computer
Editor: Addison-Wesley
Date: 1989
Content: <Text>

❑ Thuộc tính và nội dung

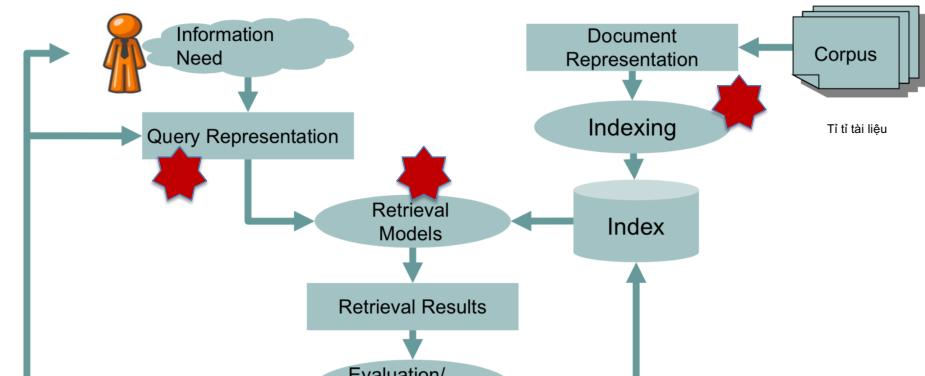
- ❑ Tìm kiếm theo thuộc tính : CSDL
- ❑ Tìm kiếm theo nội dung : IR



Các cách tiếp cận có thể

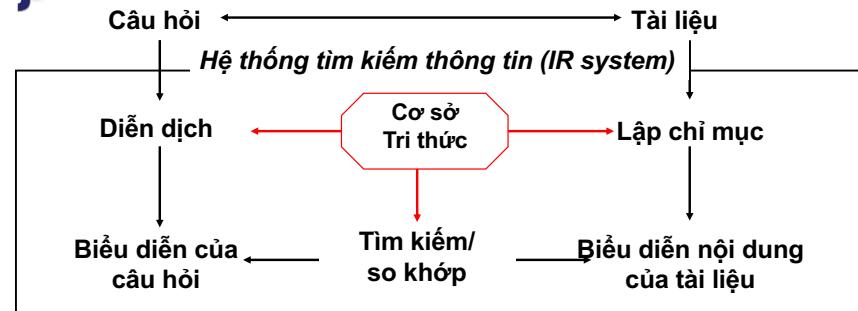
- ❑ **So khớp chuỗi** (so khớp tuyển tính chuỗi ký tự trong nội dung)
 - ❑ Chậm
 - ❑ Khó cải tiến
 - ❑ For i = 1 → 1tì: Nếu tài liệu Di chứa câu truy vấn → return Di
- ❑ ***** Lập chỉ mục** (chọn **đặc trưng** (chỉ mục) biểu diễn cho nội dung)
 - ❑ D = “this is a **computer**”
 - ❑ Nhanh
 - ❑ Linh hoạt trong việc cải tiến

Mô hình tổng quát của IR





Mô hình tổng quát của IR



Các vấn đề trong IR

- * **Lập chỉ mục cho tài liệu và câu truy vấn**
 - Làm thế nào để biểu diễn tốt nhất nội dung của chúng
- * **So khớp tài liệu và câu truy vấn**
- * **Đánh giá hệ thống:**
 - Hệ thống tốt ra sao ?
 - Chỉ trả về các tài liệu thật sự liên quan (precision)
 - Trả về tất cả tài liệu liên quan (recall)



Định nghĩa hình thức

- Một mô hình tìm kiếm thông tin là một bộ bốn $\langle D, Q, F, R(q_i, d_j) \rangle$
 - **D**: Tập các biểu diễn logic của các tài liệu trong tập dữ liệu
 - **Q**: tập hợp các biểu diễn logic cho nhu cầu thông tin của người dùng
 - **F**: Khung mô hình biểu diễn tài liệu, câu truy vấn và quan hệ giữa chúng
 - **R(q_i, d_j)**: hàm sắp xếp ánh xạ độ liên quan giữa câu truy vấn và tài liệu bằng một con số



Lập chỉ mục cho tài liệu

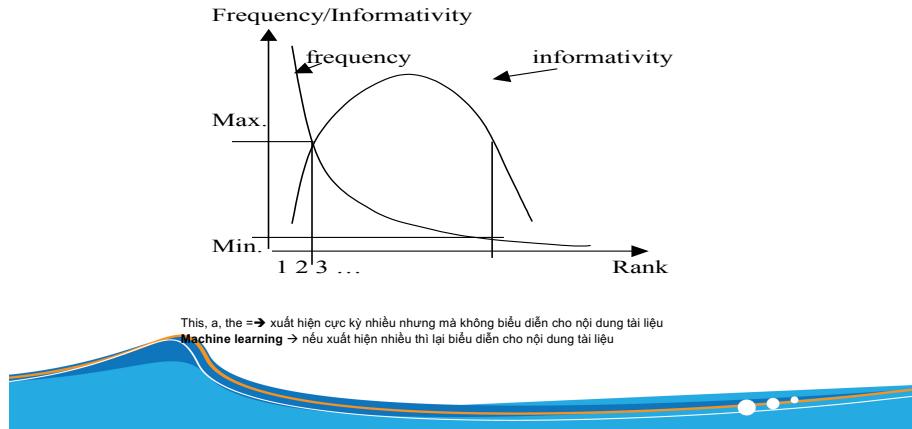
- Mục tiêu : tìm ra các “ý” quan trọng và tạo một biểu diễn trong
- **Các yếu tố cần xem xét**
 - Biểu diễn ý chính xác
 - Bao phủ nội dung
 - Máy tính có thể thao tác dễ dàng
- **Cái gì biểu diễn tốt nhất nội dung**
 - Chuỗi ký tự (char. String)
 - Từ (**Word**) *:
 - Trong tiếng Việt một từ được cấu tạo từ các tiếng: “học_sinh” là 1 từ , gồm 2 tiếng học, sinh
 - Ngữ (Phrase) *
 - Khái niệm (Concept) *





Chọn từ khóa và trọng số

- Làm thế nào để chọn từ khóa quan trọng
 - Phương pháp đơn giản : chọn từ có tần suất ở giữa



Phương pháp trọng số : tf*idf

- tf (term frequency) :
 - tần số xuất hiện của mục từ (term) trong tài liệu
 - Độ quan trọng của từ trong tài liệu
- df (Document frequency) :
 - số tài liệu chứa mục từ
 - Phân bố của từ trong kho tài liệu
 - Df càng lớn thì số tài liệu chứa từ đó càng nhiều
- Idf (Inverse document frequency):
 - Mục từ có là đặc trưng hay phổ biến: IDF càng lớn thì từ càng đặc trưng



Một vài lược đồ tf*idf

- | | |
|--|------------------------------------|
| <input type="checkbox"/> $tf(t, D) = freq(t, D)$ | $idf(t) = \log(N/n)$ |
| <input type="checkbox"/> $tf(t, D) = \log[freq(t, D)]$ | $n = \#docs \text{ containing } t$ |
| <input type="checkbox"/> $tf(t, D) = \log[freq(t, D)] + 1$ | $N = \#docs \text{ in corpus}$ |
| <input type="checkbox"/> $tf(t, D) = freq(t, D) / \text{Max}[f(t, d)]$ | |

$$\text{weight}(t, D) = tf(t, D) * idf(t)$$

- Chuẩn hóa (Normalization): Cosine normalization, /max,

...

t là 1 ý/term: từ / phrase, concept ...



Kết quả của lập chỉ mục

- Mỗi tài liệu được biểu diễn bằng một tập hợp các từ khóa có trọng số (weighted terms):
 - $D_1 \rightarrow \{(t_1, w_1), (t_2, w_2), \dots\}$
 - v.d. $D_1 \rightarrow \{(\text{comput}, 0.2), (\text{architect}, 0.3), \dots\}$
 - $D_2 \rightarrow \{(\text{comput}, 0.1), (\text{network}, 0.5), \dots\}$
 - $D_3 \rightarrow$

Tập tin nghịch đảo (Inverted file):

- tập tin nghịch đảo được sử dụng trong truy tìm nhằm tăng hiệu năng → dễ tìm kiếm nhanh



comput → {(D1, 0.2), (D2, 0.1), ...}

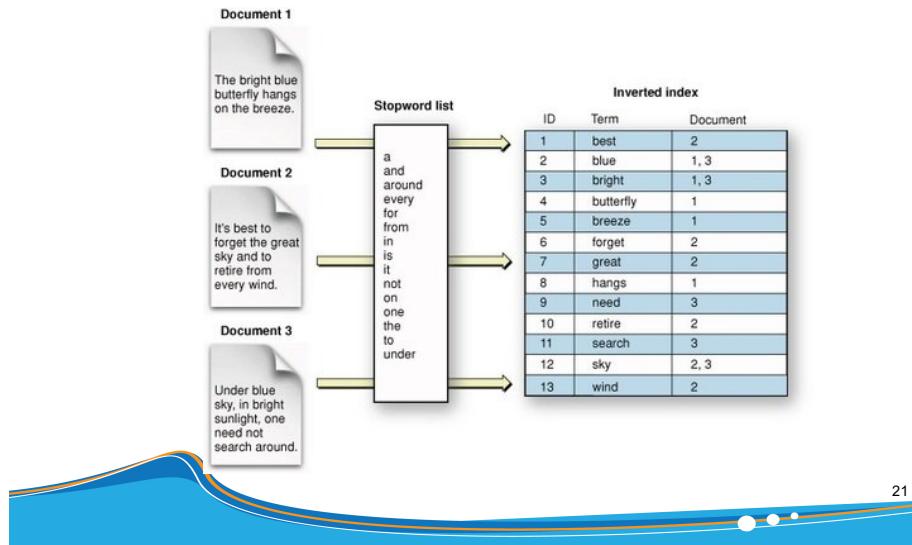
architect → {(D1, 0.3)}

network → {(D2, 0.5)}

Tìm tất cả các tài liệu chứa từ khóa "comput"

• 20

Tập tin nghịch đảo



21

Một số trường hợp

- **Câu truy vấn 1 từ:**
 - Các tài liệu có chứa từ hỏi sẽ được trả về
 - Tìm các tài liệu có chứa từ hỏi trong tập tin nghịch đảo
 - Sắp xếp theo thứ tự giảm dần của trọng số của từ hỏi trong tài liệu

Tìm ra tất cả các tài liệu chứa từ khoá "network"

- Câu truy vấn nhiều từ?
 - Tổ hợp nhiều danh sách
 - Làm thế nào để diễn dịch trọng số? => (IR model)

Tìm ra tất cả các tài liệu chứa từ khoá "network computer"



23

Truy tìm (Retrieval)

- Các vấn đề trong truy tìm
 - Mô hình truy tìm
 - Tài liệu được biểu diễn như thế nào với các mục từ đã được chọn
 - Làm thế nào để **so sánh biểu diễn của tài liệu và câu hỏi**, làm thế nào đánh giá độ liên quan
 - Các kỹ thuật cài đặt:



Mô hình truy tìm (IR models)

- Mô hình tính điểm khớp (Matching score)
 - Tài liệu D = tập hợp các từ có trọng số (tfidf)
 - Câu truy vấn Q = tập hợp các từ không trọng số

$$R(D, Q) = \sum_i w(t_i, D)$$

ở đây t_i is in Q.



24

Các mô hình căn bản của IR

Mô hình boolean

Mô hình không gian vector

*** BM25



KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



Mô hình Boolean – Cài đặt

⌘ Document Corpus

⌘ $d_1 = \text{Big cats are nice and funny}$

⌘ $d_2 = \text{small dogs are better than big dogs}$

⌘ $d_3 = \text{small cats are afraid of small dogs}$

⌘ $d_4 = \text{Big cats are not afraid of small dogs}$

⌘ $d_5 = \text{funny cats are not afraid of small dogs}$

27



Mô hình Boolean

□ Tài liệu = biểu thức Logic “và” của các từ khóa

□ Câu truy vấn = biểu thức Bool của các từ khóa

▫ $R(D, Q) = D \rightarrow Q$

▫ Nếu $D \rightarrow Q$ đúng thì $R(D, Q) = 1$, ngược lại 0

□ Ví dụ:

▫ $D = t_1 \wedge t_2 \wedge \dots \wedge t_n$

▫ $Q = (t_1 \wedge t_2) \vee (t_3 \wedge \neg t_4)$

▪ $D \rightarrow Q$, như vậy $R(D, Q) = 1$.

$(t_1 \wedge t_2 \wedge \dots \wedge t_n) \rightarrow ((t_1 \wedge t_2) \vee (t_3 \wedge \neg t_4))$

Biểu thức này cho ra chân trị T, vì vậy $R(D, Q) = 1$

□ Vấn đề:

▫ R thì hoặc 1 hoặc 0 (tập tài liệu trả về không được sắp xếp)

▫ Hoặc rất nhiều tài liệu hoặc rất ít

▫ Người sử dụng khó biểu diễn câu truy vấn dưới dạng biểu thức bool

26



Mô hình Boolean – Cài đặt

	d_1	d_2	d_3	d_4	d_5
big	1	1	0	1	0
cat	1	0	1	1	1
nice	1	0	0	0	0
Funny	1	0	0	0	1
Small	0	1	1	1	1
Dog	0	1	1	1	1
Better	0	1	0	0	0
than	0	1	0	0	0
afraid	0	0	1	1	1
not	0	0	0	1	1

⌘ Matrix element (t, d) is

1 if the document in column d contain the term in row t ,
0 otherwise.

$\text{funny} \wedge \text{dog} = D_{\text{dog}} \cap D_{\text{funny}} = \{d_2, d_3, d_4, d_5\} \cap \{d_1, d_5\} = \{d_5\}$

28



Mô hình Boolean mở rộng (để sắp xếp tài liệu trả về)

- $D = \{\dots, (t_i, w_i), \dots\}$: từ khóa có trọng số
- Diễn dịch:
 - D là thành viên của lớp t_i có độ w_i .
 - Định nghĩa theo logic mờ: $\mu_{ti}(D) = w_i$

Một giải pháp đánh giá:

$$\begin{aligned} R(D, t_i) &= \mu_{ti}(D); \\ R(D, Q_1 \wedge Q_2) &= \min(R(D, Q_1), R(D, Q_2)); \\ R(D, Q_1 \vee Q_2) &= \max(R(D, Q_1), R(D, Q_2)); \\ R(D, \neg Q_1) &= 1 - R(D, Q_1). \end{aligned}$$

29



Mô hình không gian vector

- Không gian vector = tất cả các từ khóa đã nhận biết
 $\langle t_1, t_2, t_3, \dots, t_n \rangle$

Tài liệu

$$\begin{aligned} D &= \langle a_1, a_2, a_3, \dots, a_n \rangle \\ a_i &= \text{trọng số của } t_i \text{ trong } D \end{aligned}$$

Câu truy vấn

$$\begin{aligned} Q &= \langle b_1, b_2, b_3, \dots, b_n \rangle \\ b_i &= \text{trọng số của } t_i \text{ trong } Q \end{aligned}$$

$$R(D, Q) = \text{Sim}(D, Q)$$

30



Ma trận biểu diễn

Không gian tài liệu

	t_1	t_2	t_3	...	t_n	
D_1	a_{11}	a_{12}	a_{13}	...	a_{1n}	
D_2	a_{21}	a_{22}	a_{23}	...	a_{2n}	
D_3	a_{31}	a_{32}	a_{33}	...	a_{3n}	
D_m	a_{m1}	a_{m2}	a_{m3}	...	a_{mn}	
Q	b_1	b_2	b_3	...	b_n	

Không gian mục từ

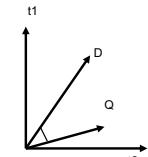
31



Một vài công thức tính độ tương tự

Tích vô hướng

$$\text{Sim}(D, Q) = D \bullet Q = \sum_i (a_i * b_i)$$



Cosine

$$\text{Sim}(D, Q) = \frac{\sum_i (a_i * b_i)}{\sqrt{\sum_i a_i^2 * \sum_i b_i^2}}$$

Dice

$$\text{Sim}(D, Q) = \frac{2 \sum_i (a_i * b_i)}{\sum_i a_i^2 + \sum_i b_i^2}$$

Jaccard

$$\text{Sim}(D, Q) = \frac{\sum_i (a_i * b_i)}{\sum_i a_i^2 + \sum_i b_i^2 - \sum_i (a_i * b_i)}$$

32



Cài đặt (không gian)

- Ma trận rất thưa: chỉ một vài trăm mục từ cho một tài liệu, và một vài từ cho câu truy vấn, trong khi đó không gian mục từ rất lớn (~100k)
- Đã được lưu trữ như sau:
 - $D_1 \rightarrow \{(t1, a1), (t2, a2), \dots\}$
 - $t1 \rightarrow \{D_1, a1, \dots\}$



Cài đặt(thời gian)

- Cài đặt VSM với tích vô hướng:
 - Cài đặt đơn giản: $O(m*n)$: m doc. & n terms
 - Cài đặt dùng tập tin nghịch đảo:

Cho câu truy vấn = $\{(t1,b1), (t2,b2)\}$:

 1. Tìm các tập hợp của các tài liệu liên quan thông qua tập tin nghịch đảo cho $t1$ and $t2$
 2. Tính điểm của các tài liệu cho mỗi mục từ có trọng số $(t1,b1) \rightarrow \{(D_1, a1 * b1), \dots\}$
 3. Tổ hợp các tập hợp và tính tổng trọng số (Σ)
 - $O(|Q|^*m)$: $|Q| << n$



Một số độ đo tương tự khác

- Cosine:

$$Sim(D, Q) = \frac{\sum_i (a_i * b_i)}{\sqrt{\sum_j a_j^2 * \sum_j b_j^2}} = \sum_i \frac{a_i}{\sqrt{\sum_j a_j^2}} \frac{b_i}{\sqrt{\sum_j b_j^2}}$$

- Sử dụng $\sqrt{\sum_j a_j^2}$ và $\sqrt{\sum_j b_j^2}$ để chuẩn hóa trọng số sau khi lập chỉ mục



BM25

- BM25 là một phương pháp xếp hạng tựa như tf-idf, được sử dụng rộng rãi trong tìm kiếm.
- Trong Web search những hàm xếp hạng này thường được sử dụng như một phần của các phương pháp tích hợp để dùng trong machine learning, xếp hạng.

[https://vi.wikipedia.org/wiki/Okapi_BM25#Ph%C6%B0%C6%A1ng_ph%C3%A1p_x%E1%BA%BFp_h%E1%BA%A1ng_\(ranking_function\)](https://vi.wikipedia.org/wiki/Okapi_BM25#Ph%C6%B0%C6%A1ng_ph%C3%A1p_x%E1%BA%BFp_h%E1%BA%A1ng_(ranking_function))



BM25

$$BM25(d_j, q_{1:N}) = \sum_{i=1}^N \left\{ IDF(q_i) \cdot \frac{TF(q_i, d_j) \cdot (k + 1)}{TF(q_i, d_j) + k \cdot (1 - b + b \cdot \frac{|d_j|}{L})} \right\}$$

Tần số xuất hiện từ khóa q_i trong d_j

Chiều dài của tài liệu d_j

Những tài liệu ngắn sẽ được ưu tiên hơn

$L = \sum_i |d_i| / N$

Chiều dài trung bình của tất cả tài liệu

i.e. calculate BM25 for each term/word in the query and sum them up
[for example BM25("president lincoln") = BM25("president") + BM25("lincoln")]

$$IDF(q_i) = \log_e \frac{N - DF(q_i) + 0.5}{DF(q_i) + 0.5}$$

Các từ xuất hiện nhiều trong các tài liệu ➔ không quan trọng

$$\text{score}(D, Q) = \sum_{i=1}^n IDF(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{avgdl}})},$$

Cách viết khác

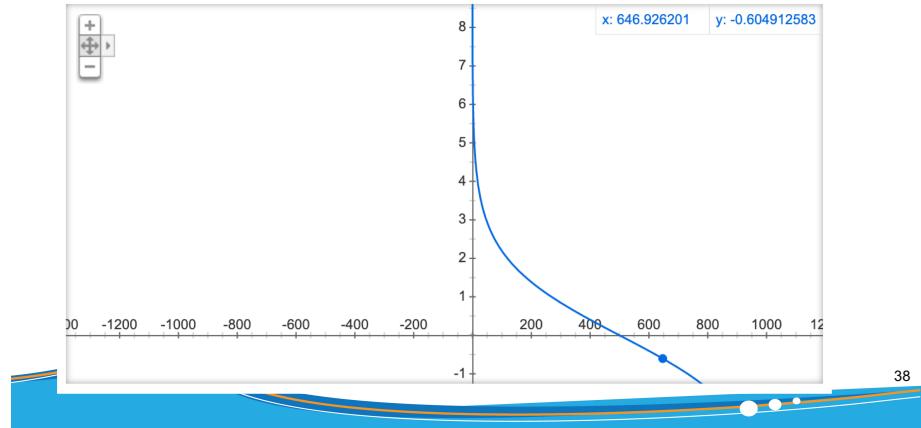
$$IDF(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5},$$

37

IDF: $y = \ln((1000-x + 0.5) / (x+0.5))$

- N = 1000, x là số tài liệu có chứa từ q_i , x càng nhỏ thì càng IDF càng lớn.

Biểu đồ cho $\ln((1000-x+0.5)/(x+0.5))$



Trình bày kết quả

- Kết quả lượng giá câu truy vấn là danh sách các tài liệu được sắp xếp theo độ tương tự của chúng với câu truy vấn.

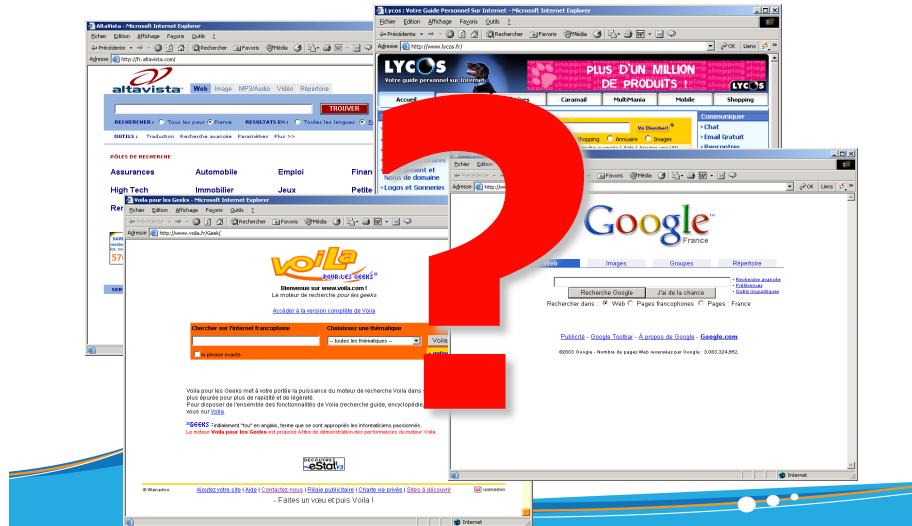
Ví dụ:

- doc1 0.67
- doc2 0.65
- doc3 0.54
- ...

39

Đánh giá hệ thống truy tìm

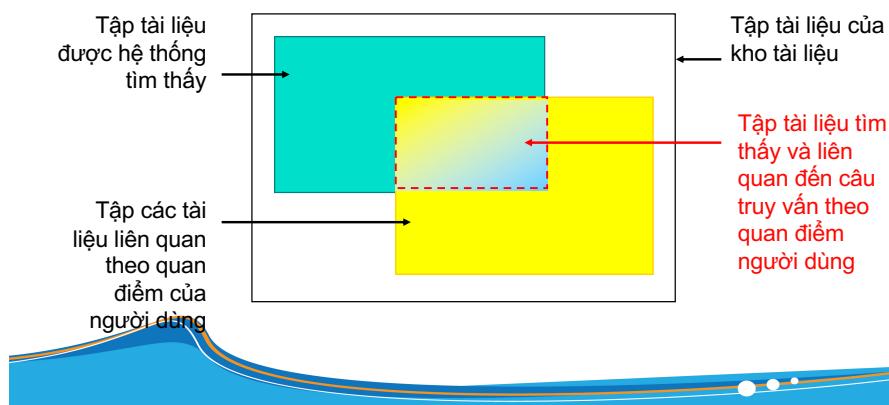
Dẫn nhập



Mục tiêu

Mục tiêu của một hệ thống tìm kiếm thông tin:

Tăng sự liên hệ giữa hai độ đo liên quan :



Khái niệm liên quan

Phân biệt hai loại liên quan :

liên quan theo đánh giá của người dùng:

- Đây là sự đáp ứng mong đợi của người dùng về các trả lời của hệ thống.
- Độ đo này đo sự thỏa mãn của người dùng so với nhu cầu thông tin của họ và với các tài liệu mà hệ thống tìm thấy.

liên quan theo hệ thống:

- Đây là sự liên quan của tài liệu với câu truy vấn theo đánh giá của hệ thống
- Tất cả tài liệu trả về đều được hệ thống đánh giá là liên quan nhiều hay ít đến câu truy vấn.



Các tiêu chí đánh giá

Cho **S** là tập các tài liệu được tìm thấy (liên quan theo hệ thống) – System.

Cho **U** là tập các tài liệu liên quan theo đánh giá của người dùng - Ground Truth

Hai tiêu chí đánh giá là :

- độ chính xác (precision)
- độ bao phủ (recall)



Độ chính xác

- Đây là sự tương ứng giữa số tài liệu mà hệ thống tìm thấy có liên quan đến câu truy vấn theo người dùng trên tổng số các tài liệu tìm thấy của hệ thống.

$$\text{Precision} = \frac{|S \cap U|}{|S|}$$

- Độ chính xác 100% nghĩa là tất cả các tài liệu mà hệ thống tìm thấy điều liên quan đến câu truy vấn theo người dùng.



Độ bao phủ

- Đây là đo sự tương quan giữa số tài liệu hệ thống tìm thấy được đánh giá là liên quan theo người dùng trên tổng số các tài liệu có liên quan theo người dùng.

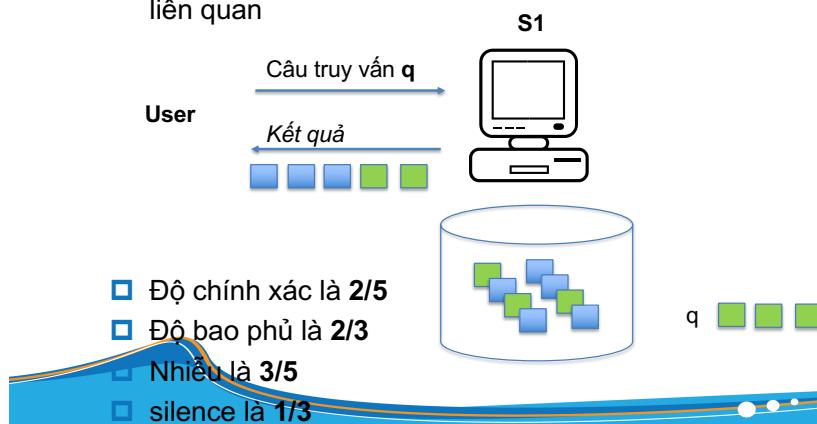
$$\text{Recall} = \frac{|S \cap U|}{|U|}$$

- Độ bao phủ là 100% có nghĩa là hệ thống tìm thấy tất cả các tài liệu liên quan.

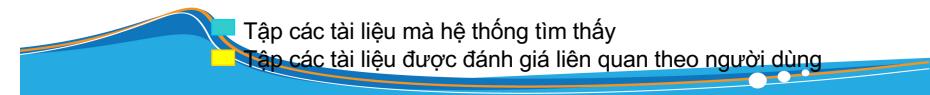
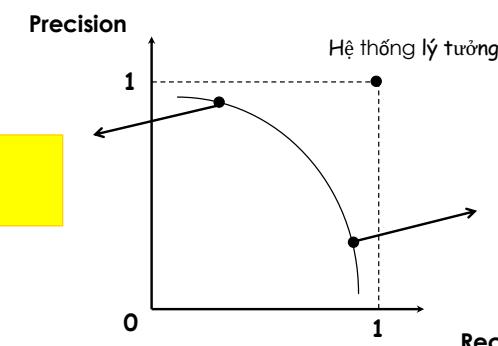


Ví dụ

- Một câu hỏi liên quan đến 3 tài liệu theo người dùng
 - S1 đã tìm thấy 5 tài liệu liên
 - Người dùng chỉ thấy có 2 tài liệu mà hệ thống tìm thấy là liên quan



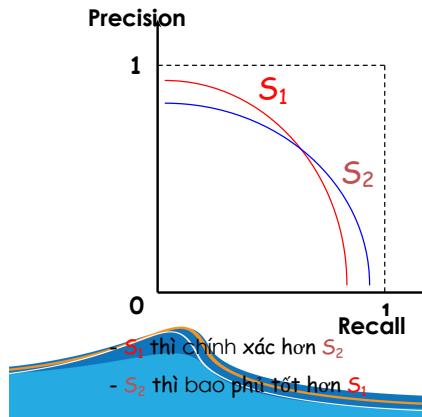
Hiệu năng



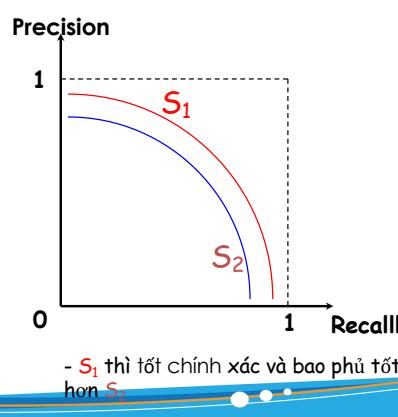
So sánh chất lượng của 2 hệ thống

Người ta so sánh thực nghiệm giữa hai hệ thống trên cùng một tập kiểm tra (test collection)

Tình huống 1



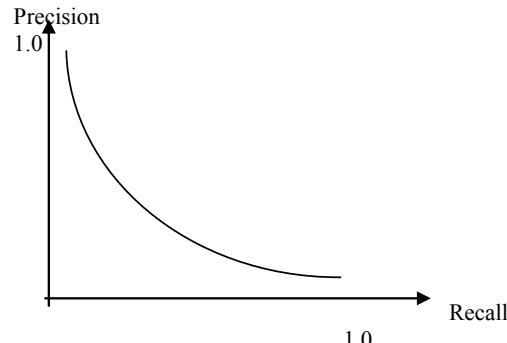
Tình huống 2



So sánh chất lượng của 2 hệ thống

- Người dùng phải tìm thấy các tài liệu liên quan càng nhanh càng tốt (tốc độ)
- Hệ thống phải sắp xếp (ranking) các tài liệu tìm thấy theo độ liên quan.
- Một hệ thống mà trả về các tài liệu đầu tiên trong danh sách phải được người dùng đánh giá là liên quan.

Dạng tổng quát của precision/recall



Precision thay đổi theo Recall

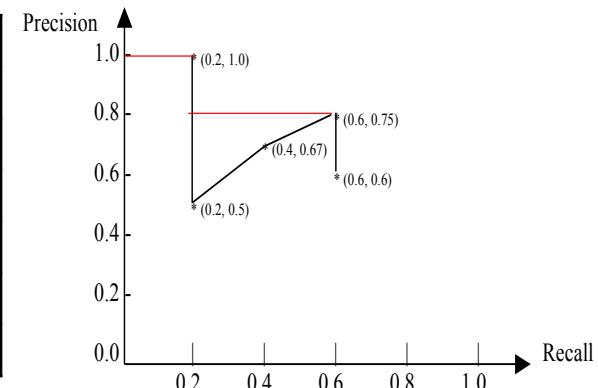
Các hệ thống không thể so sánh tại một điểm Precision/Recall

Độ chính xác trung bình (trên 11 điểm recall: 0.0, 0.1, ..., 1.0)

• 51

Một minh họa cho tính toán P/R

List	Rel?	P	R
Doc1	Y	1.00	0.2
Doc2		0.50	0.2
Doc3	Y	0.67	0.4
Doc4	Y	0.75	0.6
Doc5		0.60	0.6
...			



Assume: 5 relevant docs.

• 52

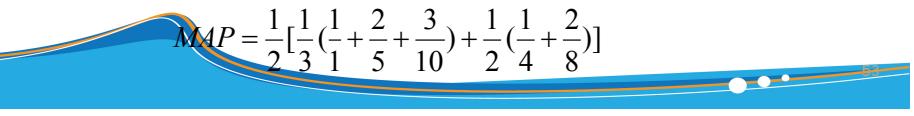


MAP (Mean Average Precision)

$$MAP = \frac{1}{n} \sum_{Q_i} \frac{1}{|R_i|} \sum_{D_j \in R_i} \frac{j}{r_{ij}}$$

- r_{ij} = thứ hạng rank of the j-th relevant document for Q_i
- $|R_i|$ = #rel. doc. for Q_i
- n = # test queries

Q1	Q2	j
1	4	1
5	8	2
10		3



$$MAP = \frac{1}{2} \left[\frac{1}{3} \left(\frac{1}{1} + \frac{2}{5} + \frac{3}{10} \right) + \frac{1}{2} \left(\frac{1}{4} + \frac{2}{8} \right) \right]$$



MAP (Mean Average Precision)

□ Dánh giá kết quả

 = relevant documents for query 1

Ranking #1                                <img alt="Icon showing two white rectangles"



Kho ngũ liệu đánh giá (Test corpus)

- So sánh các hệ thống IR khác nhau trên cùng một kho dữ liệu đánh giá
- Một kho ngũ liệu đánh bao gồm:
 - Một tập các tài liệu
 - Một tập các câu truy vấn
 - Đánh giá liên quan cho từng cặp tài liệu-truy vấn (câu trả lời mong muốn)
- Kết quả của một hệ thống sẽ được so sánh với các câu trả lời mong muốn này.

• 57



Một ví dụ về đánh giá (SMART)

Run number:	1	2	Average precision for all points
Num_queries:	52	52	11-pt Avg: 0.2859 0.3092
Total number of documents over all queries			
Retrieved:	780	780	% Change: 8.2
Relevant:	796	796	
Rel_ret:	246	229	
Recall - Precision Averages:			
at 0.00	0.7695	0.7894	
at 0.10	0.6618	0.6449	
at 0.20	0.5019	0.5090	
at 0.30	0.3745	0.3702	
at 0.40	0.2249	0.3070	
at 0.50	0.1797	0.2104	
at 0.60	0.1143	0.1654	
at 0.70	0.0891	0.1144	
at 0.80	0.0891	0.1096	
at 0.90	0.0699	0.0904	
at 1.00	0.0699	0.0904	



Demo elastic

- Basic usages with terminal:
 - Install
 - Create Index
 - Indexing data
 - Query
- Elastic with Python:
 - Connect to elastic search
 - Query with python
- Write API with Flask
- Write Simple Web App with Flask



Xây dựng hệ tìm kiếm bằng elastic search

<https://www.elastic.co/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables>



KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

• 57

60

Download Elastic Search

```
! wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.5.4.tar.gz  
! wget https://artifacts.elastic.co/downloads/kibana/kibana-6.5.4-darwin-x86_64.tar.gz
```

Run 1

Terminal

```
sonnt:bin sonnguyen$ cd /Users/sonnguyen/Downloads/elasticsearch/elasticsearch-6.5.4/bin  
sonnt:bin sonnguyen$ ./elasticsearch & |
```



Hướng tiếp cận mới

Dense Passage Retrieval for Open-Domain Question Answering

**Vladimir Karpukhin, Barlas Oğuz*, Sewon Min[†], Patrick Lewis,
Ledell Wu, Sergey Edunov, Danqi Chen[‡], Wen-tau Yih**
Facebook AI [†]University of Washington [‡]Princeton University
dk, barlaso, plewis, ledell, edunov, scottiyih}@fb.com
sewon@cs.washington.edu
dangic@cs.princeton.edu

Condenser: a Pre-training Architecture for Dense Retrieval

Luyu Gao and Jamie Callan
Language Technologies Institute
Carnegie Mellon University
{luyug, callan}@cs.cmu.edu



Dense Passage Retrieval

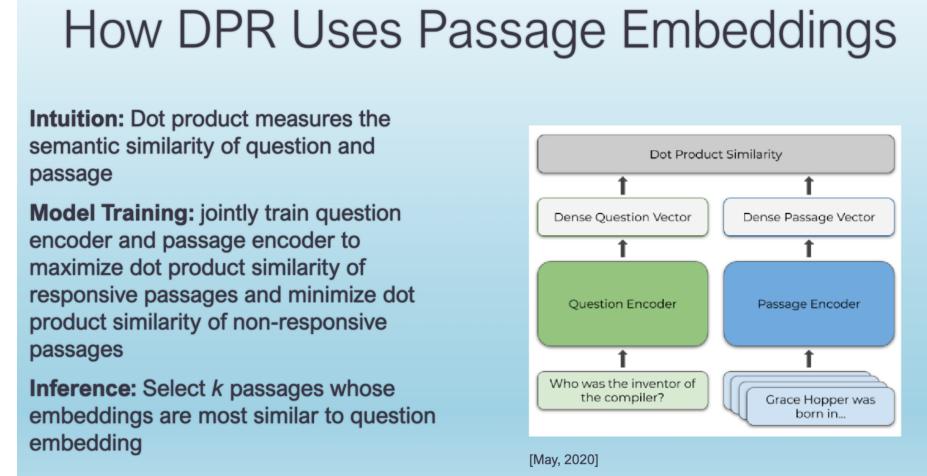
Dense Passage Retrieval for Open-Domain Question Answering

Vladimir Karpukhin¹, Barlas Oğuz², Sewon Min¹, Patrick Lewis,
Ledell Wu¹, Sergey Edunov¹, Danqi Chen¹, Wen-tau Yih¹
Facebook AI¹ University of Washington² Princeton University
{vladk, barlaso, plewis, ledell, edunov, scottiyih}@fb.com
sewon@cs.washington.edu
dchen@cs.princeton.edu

Abstract

Open-domain question answering relies on efficient passage retrieval to select candidate contexts, where traditional sparse vector space models, such as TF-IDF or BM25, are the de facto standard. However, the performance of retrieval can be practically improved using dense representations alone, where embeddings are learned from millions of questions and passages by a simple dual encoder framework. When evaluated on a

- ❖ Uses passage embeddings to semantically match an embedded query to the most responsive passages
 - ❖ Still a two-stage retriever-reader pipeline. However,
 - ❖ The retriever uses semantic similarity
 - ❖ The reader uses the embedded query to identify the answer text from retrieved passages



DPR Results

- ❖ DPR achieves SOTA on 4/5 benchmarks
- ❖ Performs poorly when trained on small datasets.
 - ❖ Must be trained on more data (multiple datasets) to achieve good metrics
- ❖ Performs better on real questions harvested from server logs (WebQuestions, NaturalQuestions) than on questions formulated when answer is already known (TriviaQA, SQuAD) [Chen, 2020]

Training	Model	NQ	TriviaQA	WQ	TREC	SQuAD
Single	BM25+RFRT (Lee et al., 2019)	26.5	47.1	17.7	21.3	33.2
Single	ORQA (Lee et al., 2019)	33.3	45.0	36.4	30.1	20.2
Single	HardeM (Min et al., 2019a)	28.1	50.9	-	-	-
Single	GraphRetriever (Min et al., 2019b)	34.5	56.0	36.4	-	-
Single	PathRetriever (Asai et al., 2020)	32.6	-	-	-	56.5
Single	REALM _{Wiki} (Guu et al., 2020)	39.2	-	40.2	46.8	-
Single	REALM _{News} (Guu et al., 2020)	40.4	-	40.7	42.9	-
BM25		32.6	52.4	29.9	24.9	38.1
Single	DPR	41.5	56.8	34.6	25.9	29.8
Single	BM25+DPR	39.0	57.0	35.2	28.0	36.7
Multi	DPR	41.5	56.8	42.4	49.4	24.1

Bài tập

- “Building a web search engine in practical”
- Kiến trúc ?
- Các vấn đề cần giải quyết ?

