



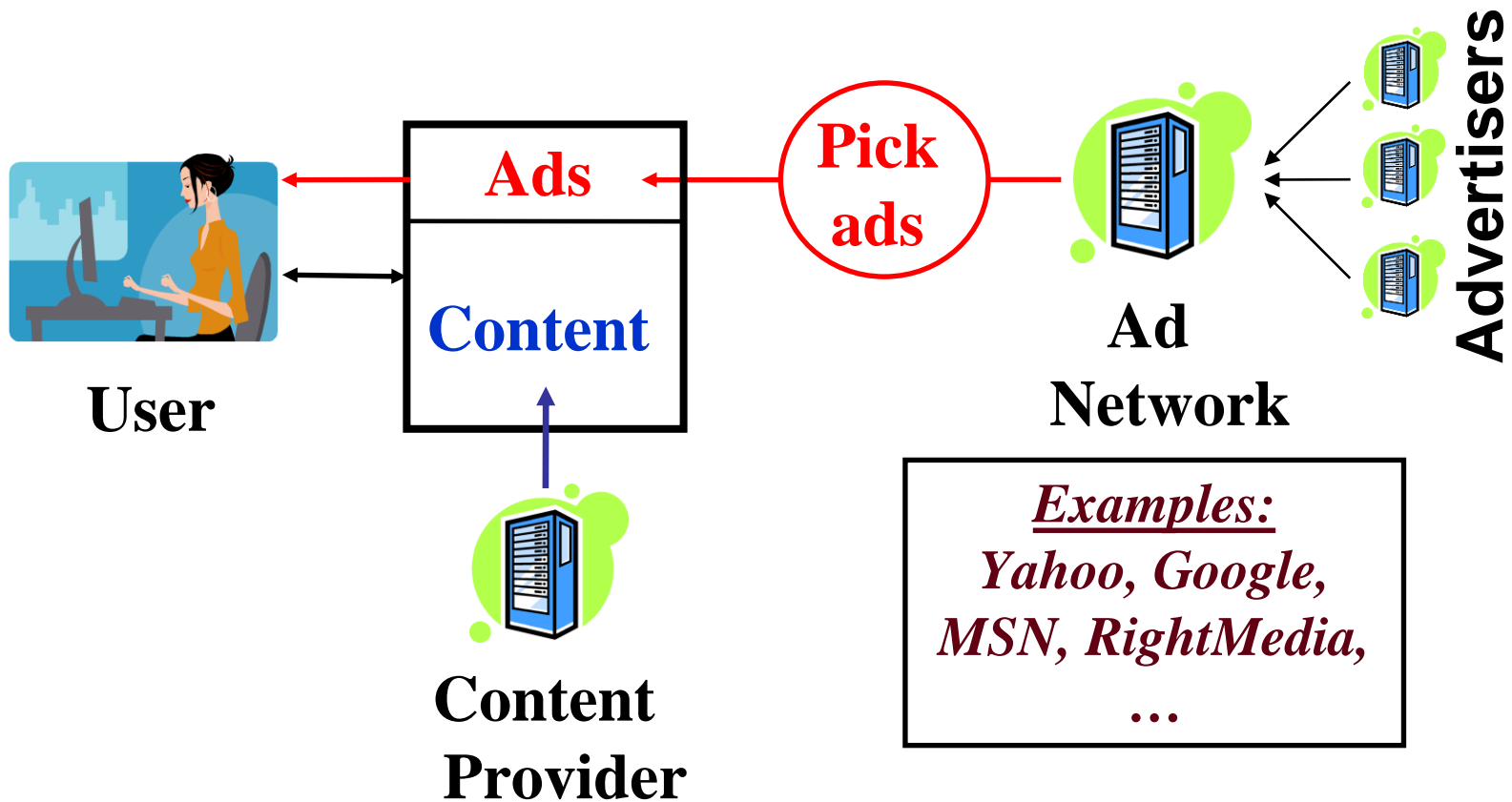
ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# Lesson 9: Online Ad & Query Mining

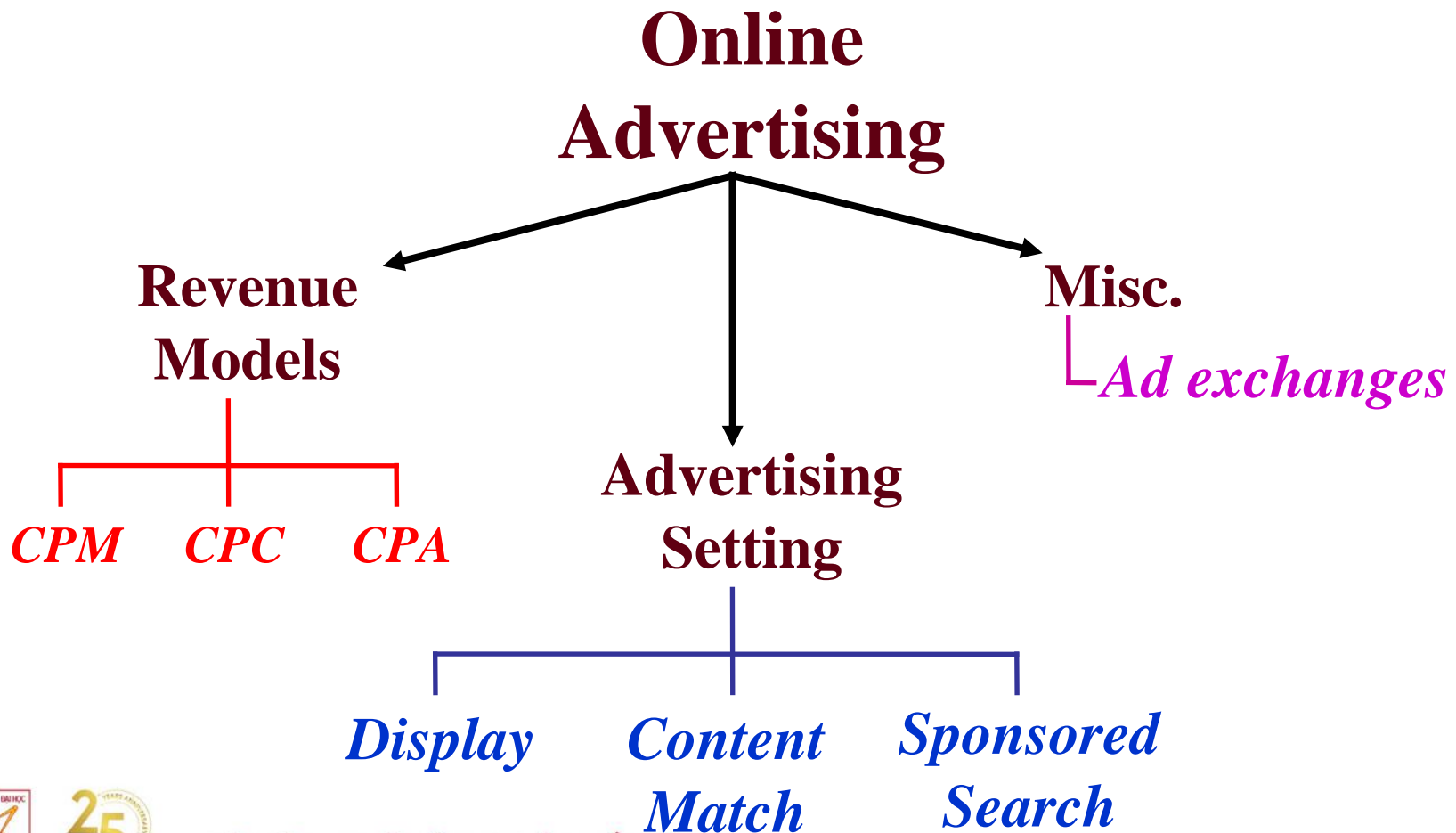
# Agenda

1. Online advertising
2. Search engine advertising
3. Query Mining

# 1. Online Advertising



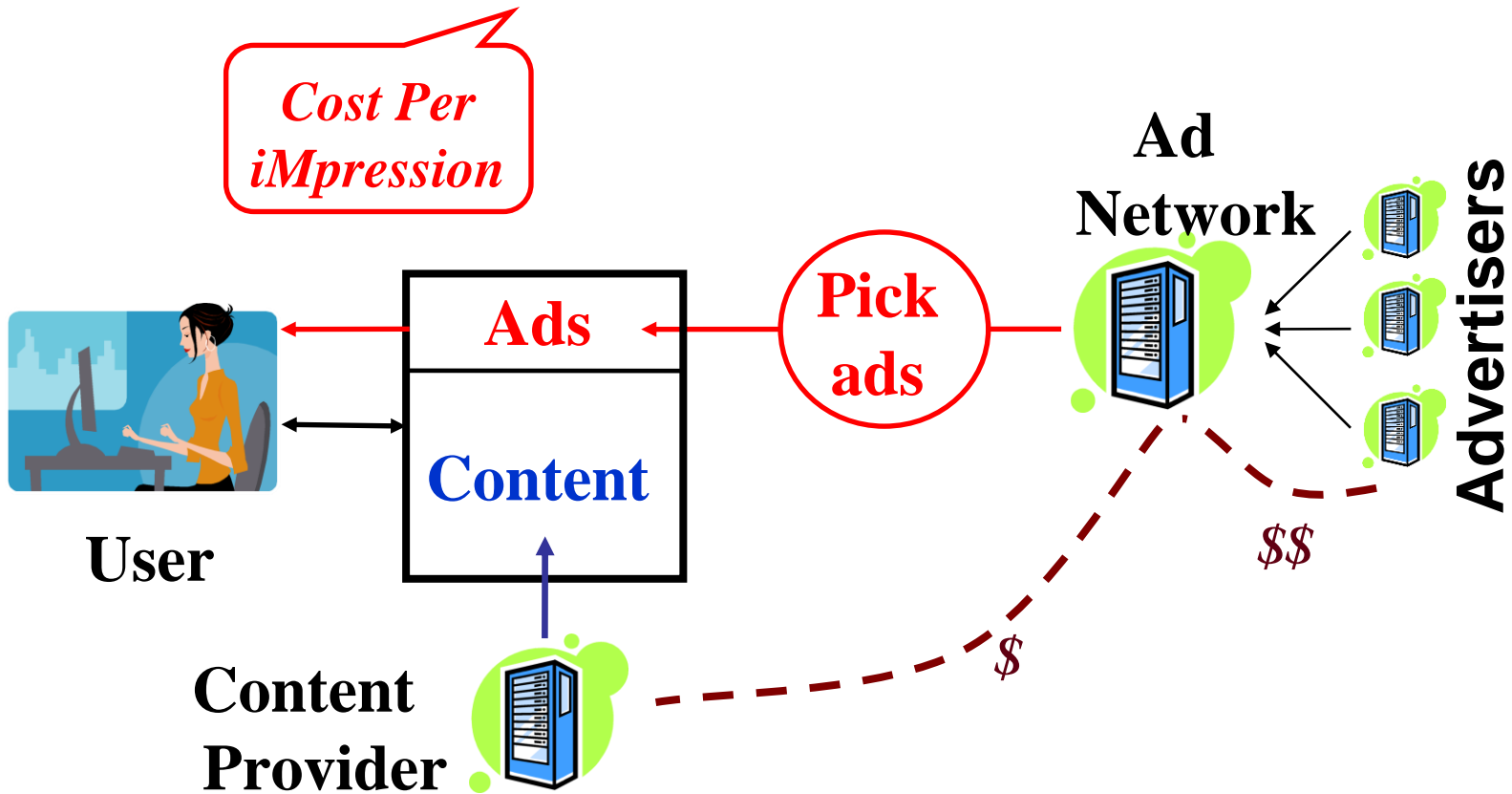
# Online advertising model



# CPM

CPM CPC CPA

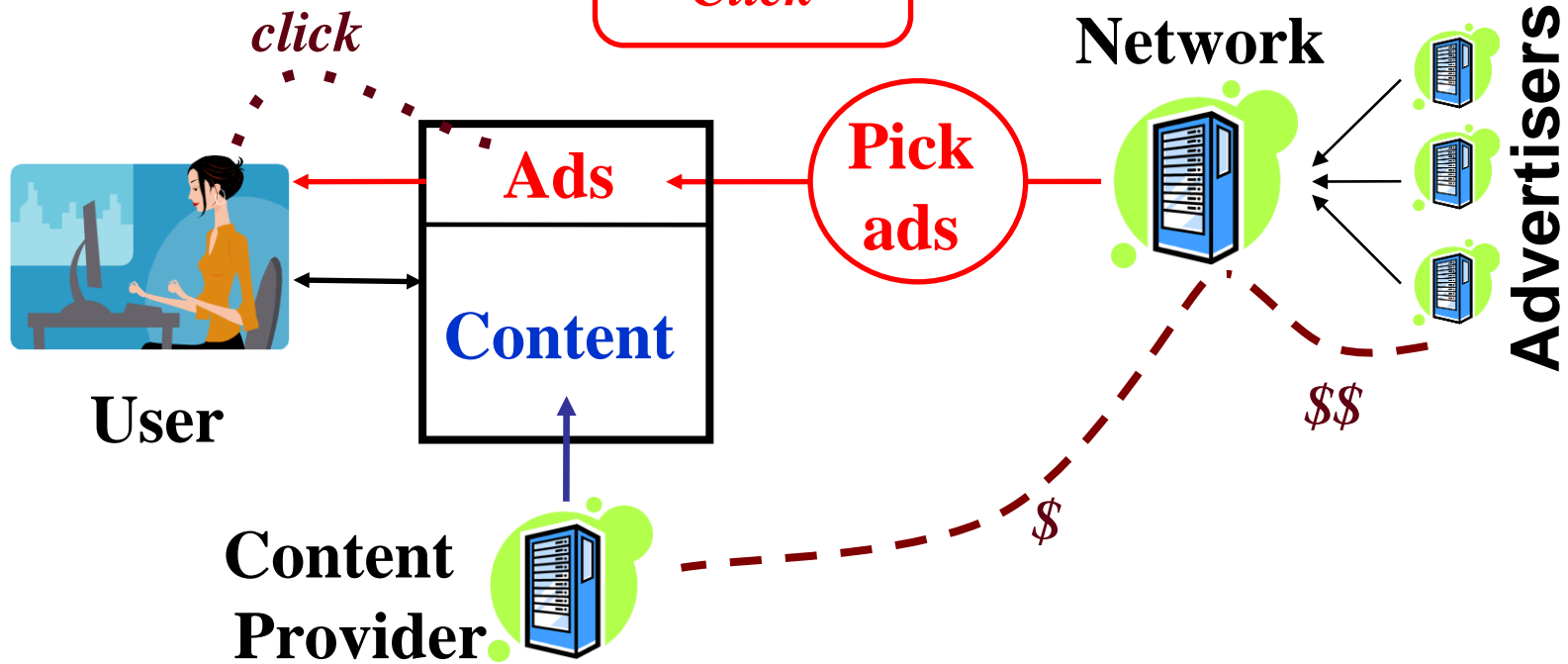
*Cost Per  
iMpression*



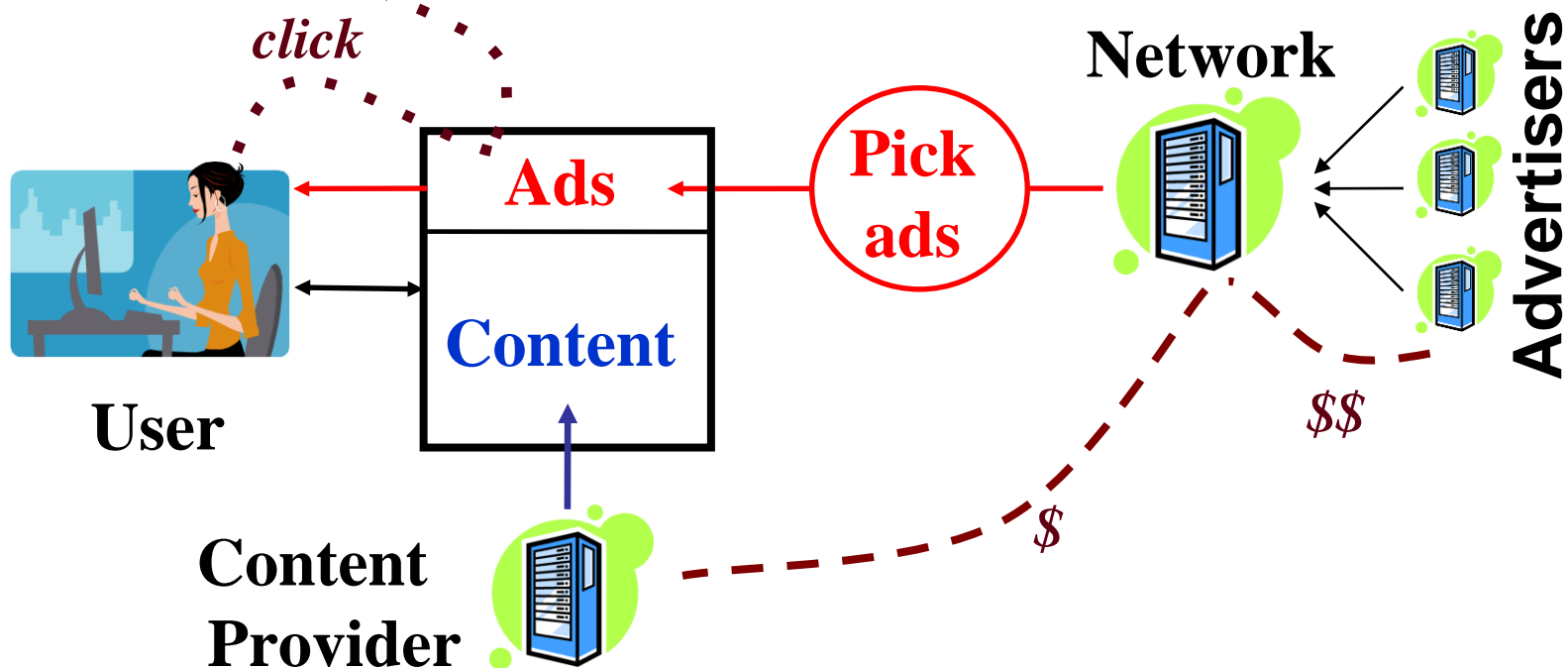
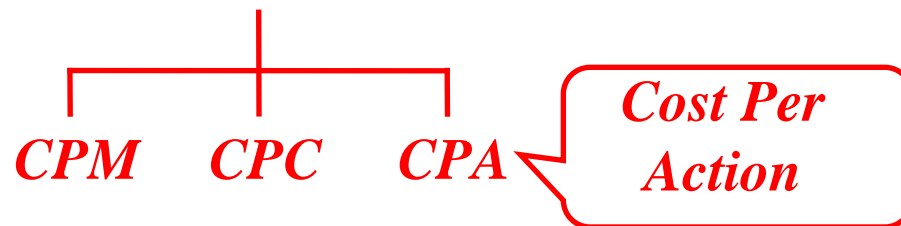
# CPC

*CPM*   *CPC*   *CPA*

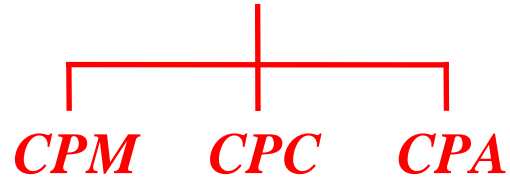
*Cost Per Click*



# CDA



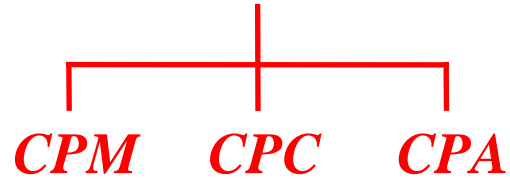
# Revenue - CPM



- Assume that an ad is shown N items at the same position
- CPM: Revenue =  $N * CPM$



# Revenue - CPC



- Assume that an ad is shown N items at the same position
- CPM: Revenue = N \* CPM
- CPC: Revenue = N \* **CTR** \* CPC

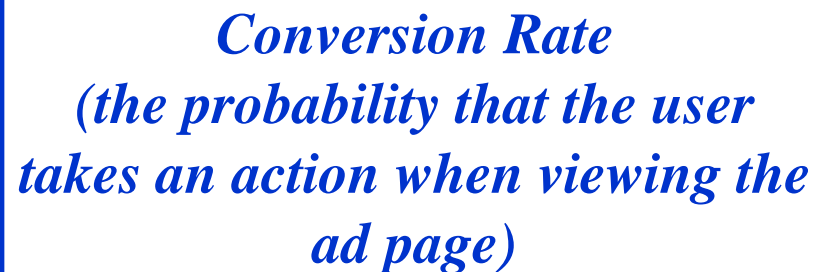
*Depends on  
the auction  
mechanism*

*Click-through Rate  
(probability of clicking  
on an ad)*

# Revenue - CPA

  
*CPM CPC CPA*

- Assume that an ad is shown N items at the same position
- CPM: Revenue = N \* CPM
- CPC: Revenue = N \* CTR \* CPC
- CPA: Revenue = N \* CTR \* **Conv. Rate** \* CPA

  
*Conversion Rate*  
*(the probability that the user*  
*takes an action when viewing the*  
*ad page)*

## 2. Search engine advertising

The screenshot shows a Mozilla Firefox browser window displaying Yahoo! search results for the query "recipe indian food". The browser's address bar shows the search URL. The search results page features a "Query" label pointing to the search bar and a "Paid Ad" label pointing to a sponsored result. The search results are divided into "Search Results" and "SPONSOR RESULTS".

**Query:** recipe indian food

**Paid Ad:** Indian Food

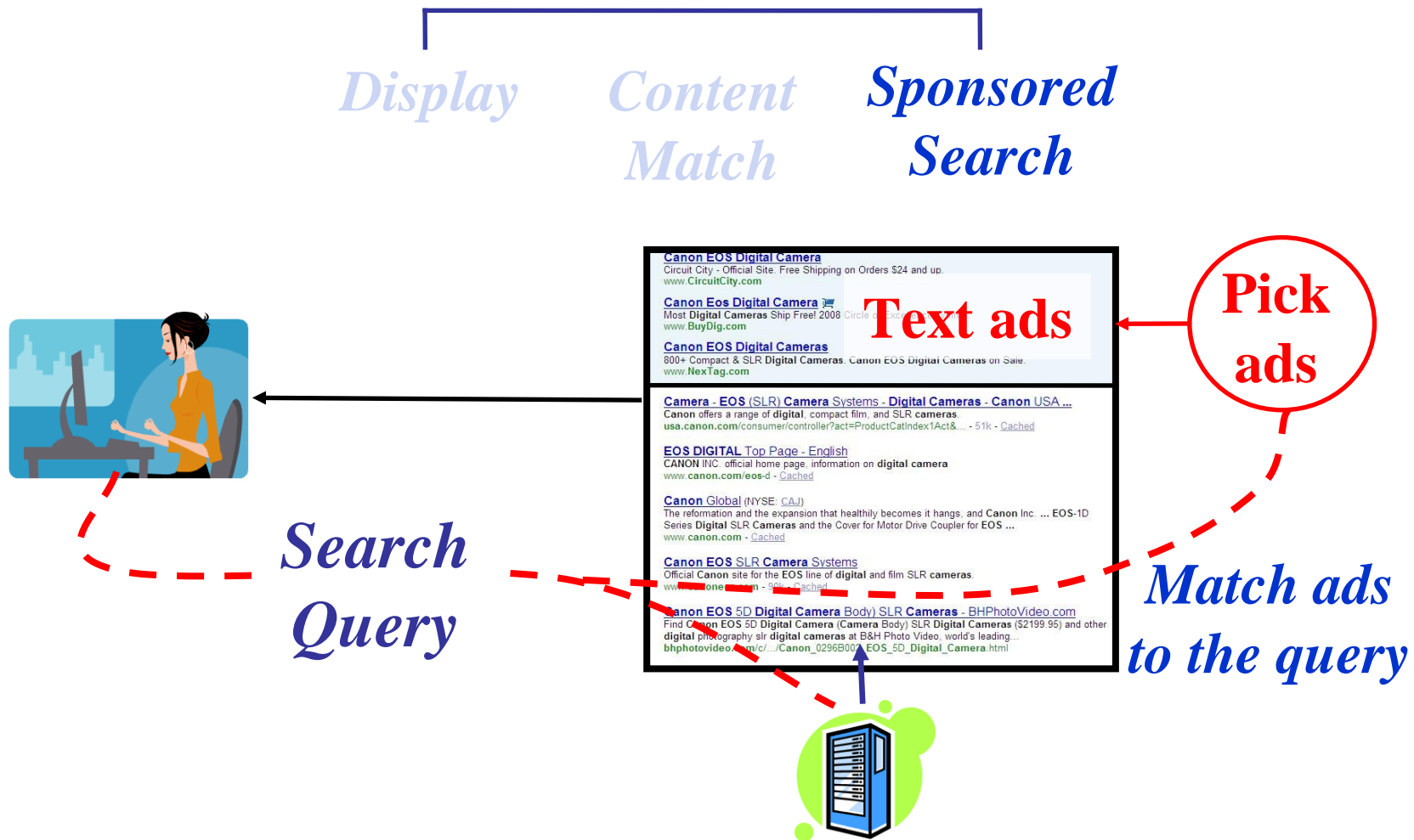
**Search Results:**

- 1. **Recipe Indian Food**  
www.MonsterMarketplace.com - Browse and compare great deals on recipe indian food.
- 2. **Indian Food**  
sanfrancisco.citysearch.com - Find great Indian restaurants in your area today. Search here.
- 3. **Indian food recipe**  
indian food recipe ... Title: Indian Food Recipe. Yield: 4 Servings. Ingredients. 1 bunch ... to the echo by: Jonathan Kandell Indian Food Recipes Put ...  
recipes.chef2chef.net/recipe-archive/43/231458.shtml - 13k - Cached - More from this site
- 4. **Recipe Gal: Indian Foods**  
Indian Recipes from Recipe Gal's Archives ... All Food Posters. Travel Posters. Indian Recipes. Indian Breads Indian Chicken Recipes ...  
www.recipegal.com/indian - 10k - Cached - More from this site
- 5. **Indian Recipes, Indian Food Recipe, South Indian Recipes, Indian ...**  
indian recipes, indian food recipe, south indian recipes, indian cooking recipes, ... Indian Recipes, Indian Food Recipe, South Indian Recipes, Indian Cooking Recipe, ...  
www.india4world.com/indian-recipe - 17k - Cached - More from this site
- 6. **Paav Bhaaji - Recipe for Paav Bhaaji - Pao Bhaaji**

**SPONSOR RESULTS:**

- Indian Food**  
Buy indian food at SHOP.COM  
Search our free shipping offers.  
www.SHOP.com
- Recipe India Food**  
Find and Compare prices on recipe india food at Smarter.com.  
www.smarter.com
- Chinese Food Recipe Books on CatalogLink**  
Find chinese food recipe books on CatalogLink.  
www.CatalogLink.com
- \$19.97 Over 500 Chinese Recipes Cookbook**  
100% Satisfaction Guaranteed  
243-Page Chinese Cookbook. Only \$19.97.

# Search engine advertising model



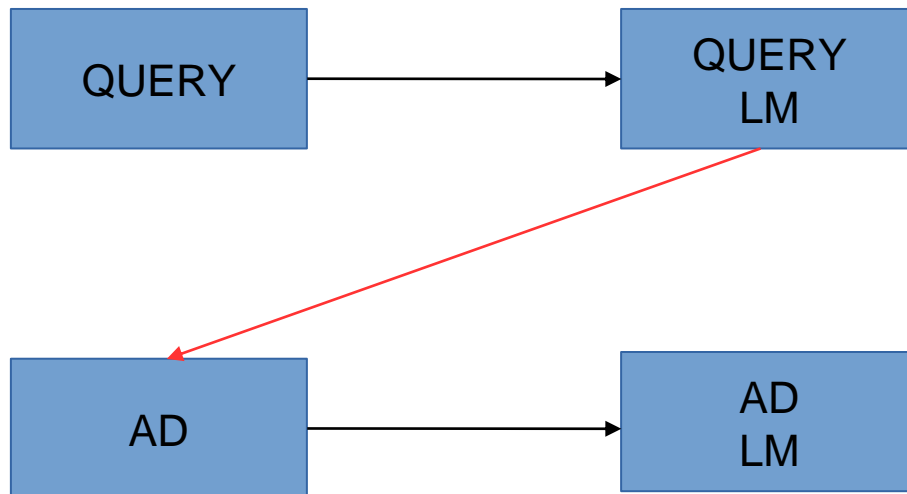
# Maximize revenue

- The problem of advertising company
- Select ads for maximum revenue
  - Match the query
  - Advertising costs
  - Ad page quality

# Scoring based on content

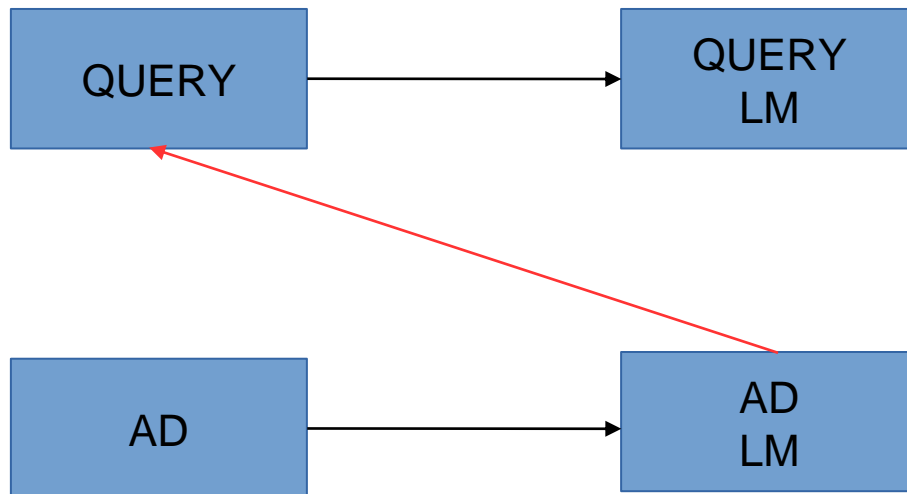
- Consider advertising like a text
- Compare query similarity to ads
- Methods
  - Vector space model
  - Language model

# Language model



$P(ad|query$   
LM)

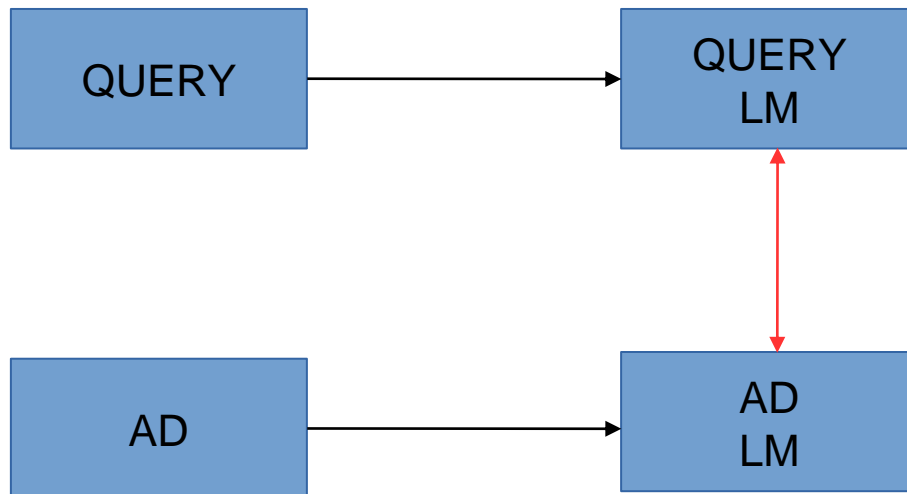
# Language model (cont.)



$P(\text{query}|\text{ad LM})$



# Language model (cont.)



$KL(ad\ LM; query\ LM)$

# Pros and Cons

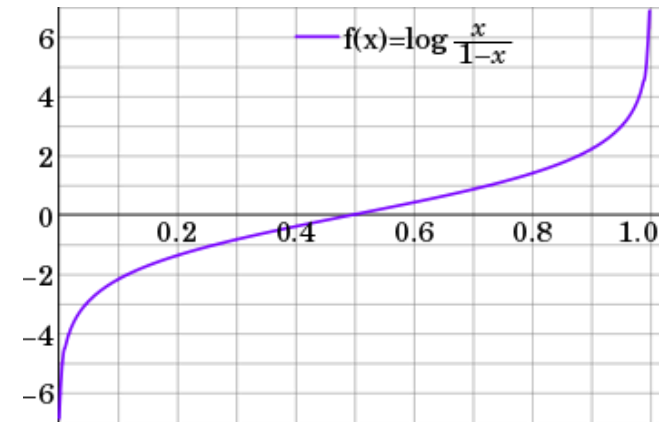
- Pros
  - Simple model
  - Suitable for short popular query
- Cons:
  - Hardly handle rare queries (long tail)
  - Hardly process in real-time
  - Not using user feedback

# Score based on user feedback

- Query set  $Q$
- Ad page set  $A$
- For each query  $q \in Q$  and ad page  $a \in A$ , compute the probability that user clicks on ad page  $\Pr(\text{click} | q, a)$
- Using user feedback to estimate probabilities

# Logistic Regression

- Representation of query and advertising content in vectors (bag of words)
- $\Pr(\text{click} | q, a) = f(\mathbf{q}, \mathbf{a}; \theta)$
- Logistic Regression:
  - Log-odds ( $\Pr(\text{click} | q, a)$ ) =  $\mathbf{q}' \mathbf{W} \mathbf{a}$
  - Estimate  $\mathbf{W}$  using user feedback as training data

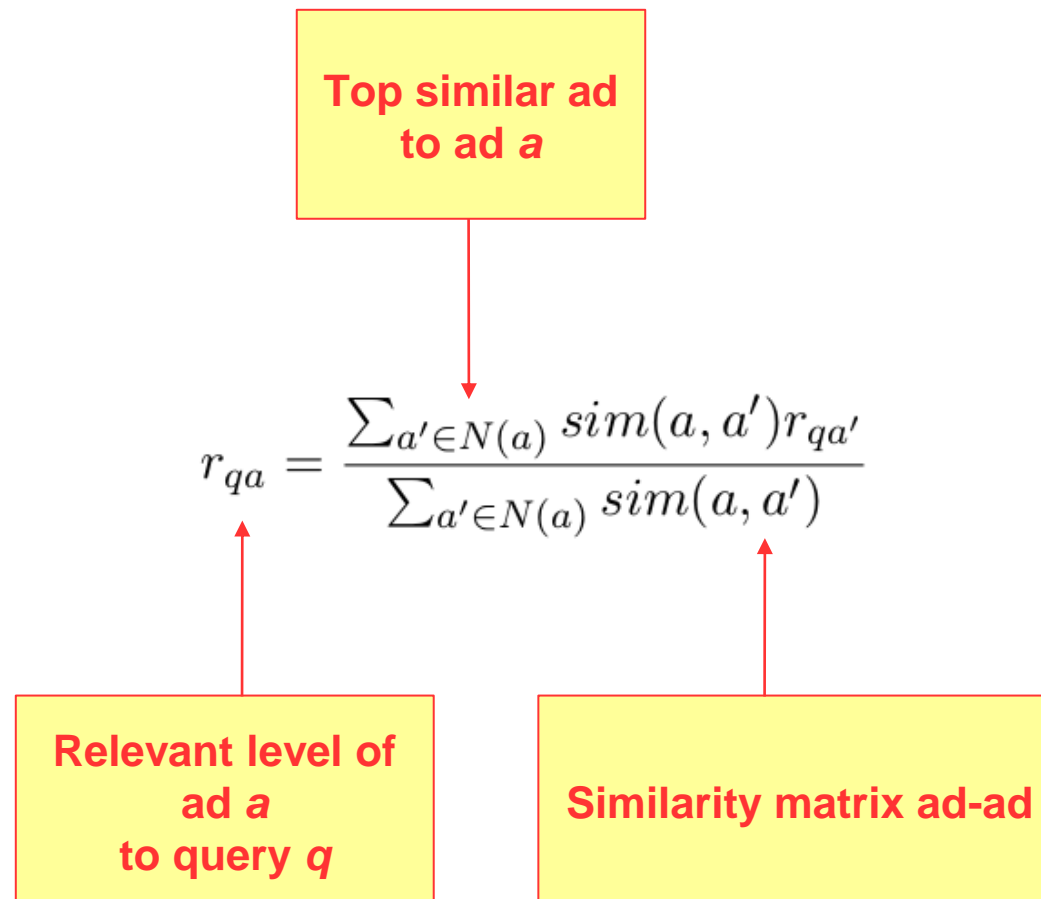


from Wikipedia

# Collaborative filtering

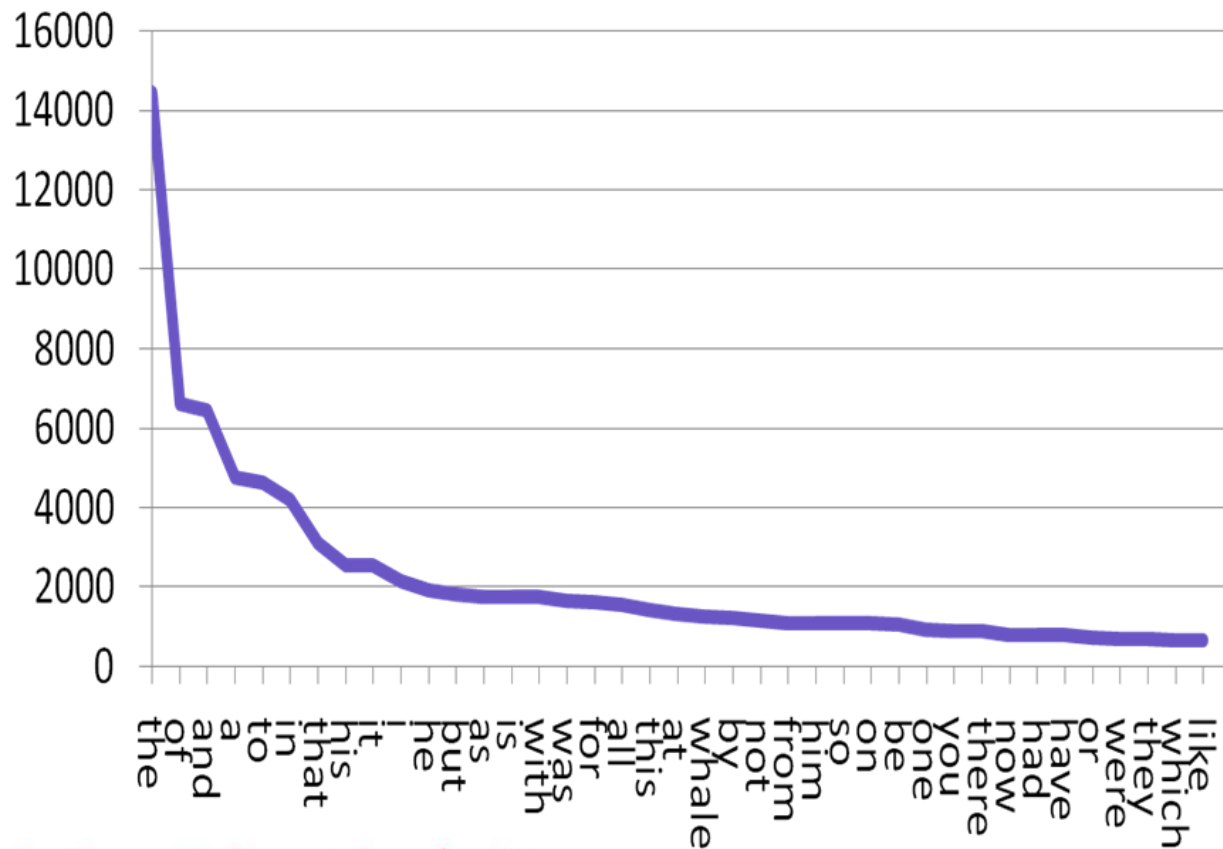
- Interactive matrix query, advertising
- Use latent user feedback (click on ad page)
- For each query  $q$  and ad page  $a$ , predict user interest
- Collaborative filtering
  - Using kNN
  - Represent ads by query to calculate similarity

# Collaborative filtering (cont.)



# 3. Query Mining

- Google: 40,000 query/s



# Query features

- A query contains an average of 2.4 words
- 21% of internet traffic comes from search engines
- User feedback
  - 50% click on first result
  - Users mostly only use the first two results



# Query features (cont.)

- Users often edit query
- Search trends shift from entertainment to e-commerce, in which product search accounts for 1/5
- The distribution of vocabulary on the query and on the website content is different → what users search for is different from what is available on the internet

# Query logging

- User information
- Query content
- List of relevant documents
- Selected documents of user

# Query preprocessing

- Identify query session
- Filter bot query
- Standardize query

# Identify query session

- Classify pairs of consecutive queries into classes :
  - Same query content but different search scope
  - Query Generalization
  - Query fine-tuning for a more precise query
  - Query detailing
  - New query content

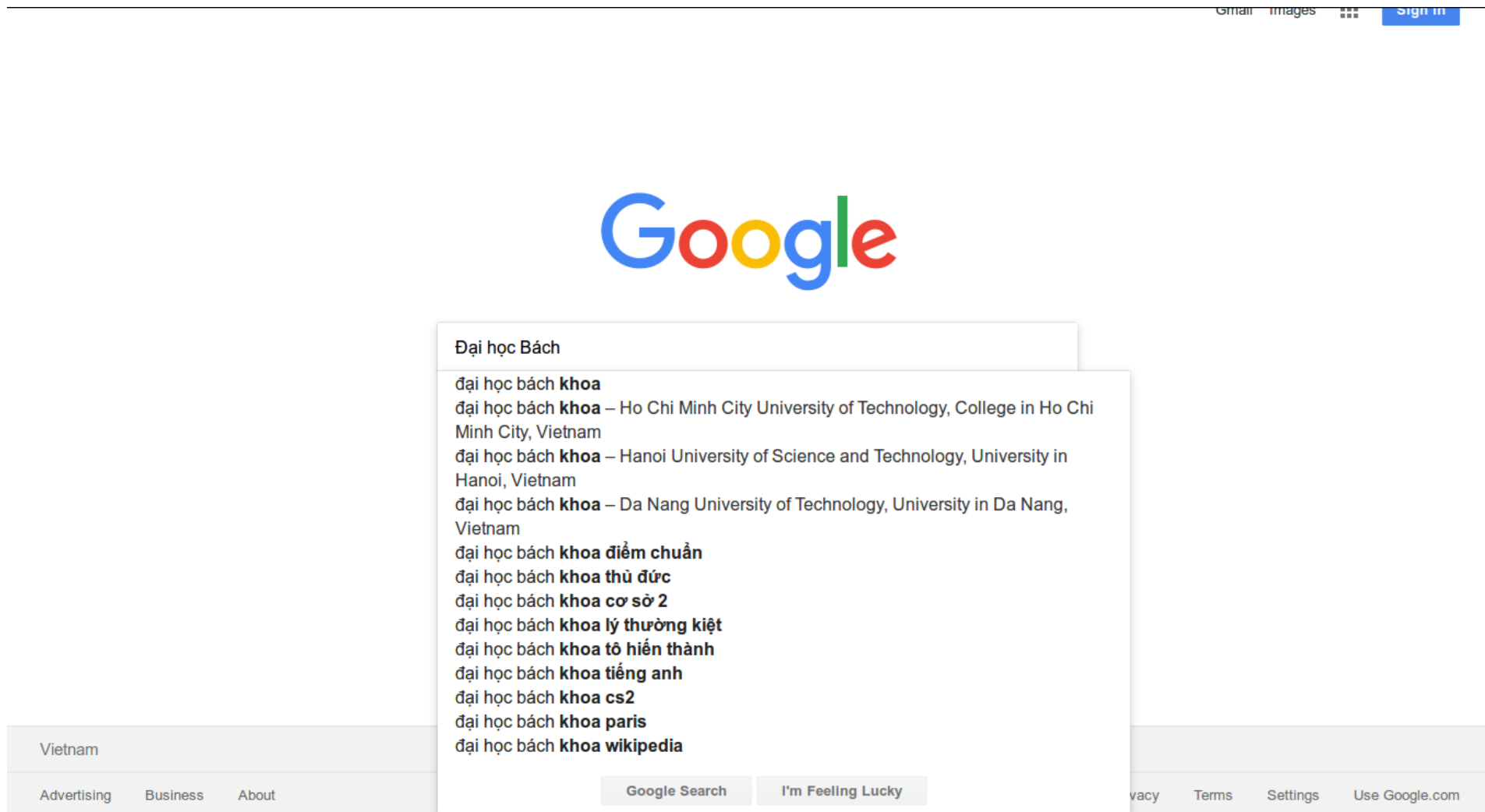
# Filter bot query

- Query generated by bot to collect search engine results
- Duplicate content
- Unusually high query rate and/or recurring query frequency

# Standardize query

- Remove stopwords
- Convert to lower case
- Standardize number
- Stemming
- For Vietnamese
  - Restore accent
  - Tokenize

# Application 1: Query suggestion



Report inappropriate predictions

# Language model

- Learn language model on query data  
$$\operatorname{argmax}_w P(w|w_0, w_1, \dots, w_{n-1}, w_n)$$
- Require large query dataset
- The basic unit of the language model
  - word (tokenize)
  - syllable
  - demisyllable ('ch', 'ang')
  - Character



# n-gram language model

- Unigram

$$P(w) = (\text{count}(w)+1) / (\sum_{w'} \text{count}(w')+V)$$

- Bigram

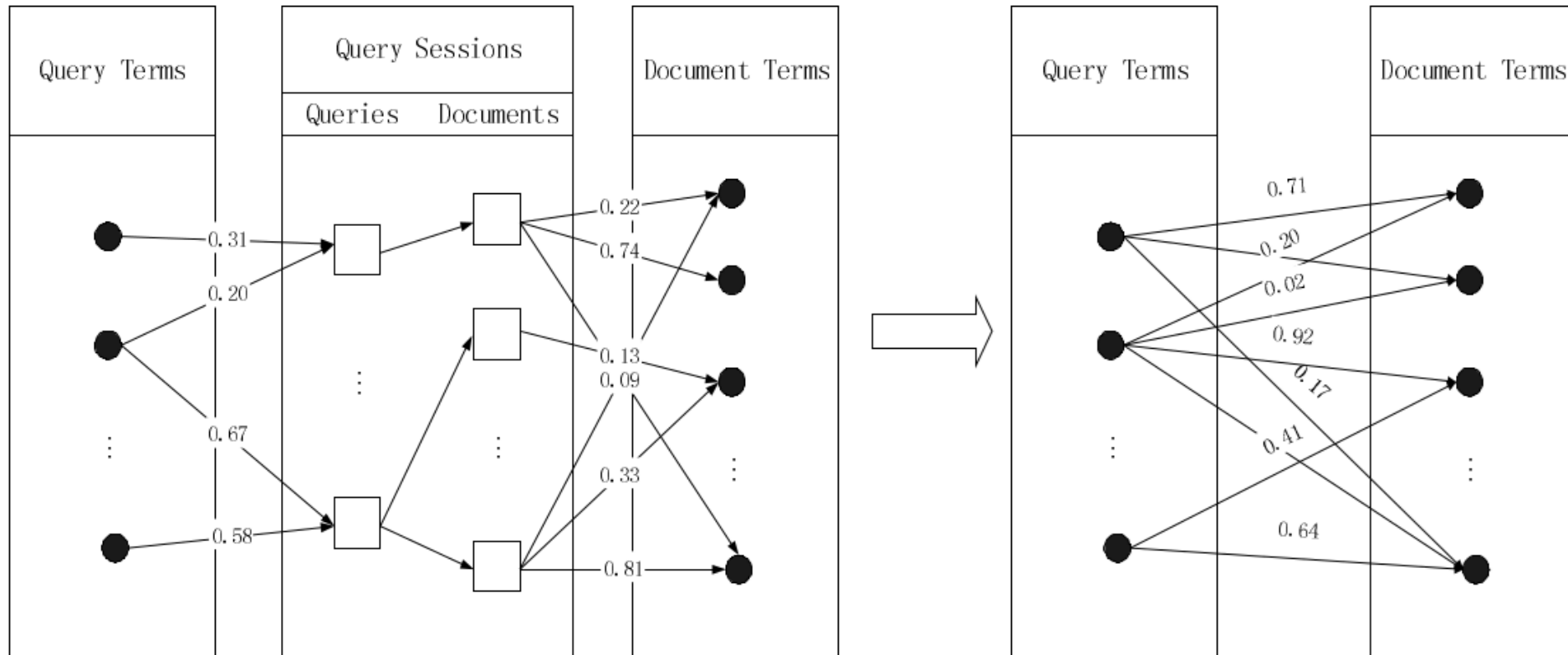
$$P(w_0, w_1) = P(w_1|w_0) * P(w_0)$$

$$P(w_1|w_0) = (\text{count}(w_0, w_1)+1) / (\sum_{w'} \text{count}(w_0, w')+V)$$

## Application 2: Extend query

- User queries often do not contain enough information
- Query expansion based solely on textual content may not meet user needs properly
  - Using user feedback
- Assumption: If a query containing one keyword leads to related documents containing another keyword, it is likely that the two keywords are related.

# Extend query model



## Extend query model (cont.)

$$\begin{aligned} P(w_j^{(d)} | w_i^{(q)}) &= \frac{P(w_j^{(d)}, w_i^{(q)})}{P(w_i^{(q)})} \\ &= \frac{\sum_{\forall D_k \in S} P(w_j^{(d)}, w_i^{(q)}, D_k)}{P(w_i^{(q)})} \\ &= \frac{\sum_{\forall D_k \in S} P(w_j^{(d)} | w_i^{(q)}, D_k) \times P(w_i^{(q)}, D_k)}{P(w_i^{(q)})} \end{aligned}$$

# Extend query model (cont.)

$$P(w_j^{(d)} | w_i^{(q)}, D_k) = P(w_j^{(d)} | D_k)$$

$$\begin{aligned} P(w_j^{(d)} | w_i^{(q)}) &= \frac{\sum_{\forall D_k \in S} P(w_j^{(d)} | D_k) \times P(D_k | w_i^{(q)}) \times P(w_i^{(q)})}{P(w_i^{(q)})} \\ &= \sum_{\forall D_k \in S} P(w_j^{(d)} | D_k) \times P(D_k | w_i^{(q)}) \end{aligned}$$

$P(w_j^{(d)} | D_k)$  : probability of  $w_j^{(d)}$  given selected  $D_k$

$P(D_k | w_i^{(q)})$  : probability of  $D_k$  to be selected if  $w_i^{(q)}$  appears in query

# Extend query model (cont.)

$$P(D_k | w_i^{(q)}) = \frac{f_{ik}^{(q)}(w_i^{(q)}, D_k)}{f^{(q)}(w_i^{(q)})}$$

$$P(w_j^{(d)} | D_k) = \frac{W_{jk}^{(d)}}{\max_{\forall t \in D_k} (W_{tk}^{(d)})}$$

$$P(w_j^{(d)} | w_i^{(q)}) = \sum_{\forall D_k \in S} (P(w_j^{(d)} | D_k) \times \frac{f_{ik}^{(q)}(w_i^{(q)}, D_k)}{f^{(q)}(w_i^{(q)})})$$

$f_{ik}^{(q)}(w_i^{(q)}, D_k)$  : number query session in which query contain  $w_i^{(q)}$  and  $D_k$  is seleted

$f^{(q)}(w_i^{(q)})$  : number of query session in which query contain  $w_i^{(q)}$

$W_{jk}^{(d)}$  : Weight of  $w_j^{(d)}$  in document  $D_k$

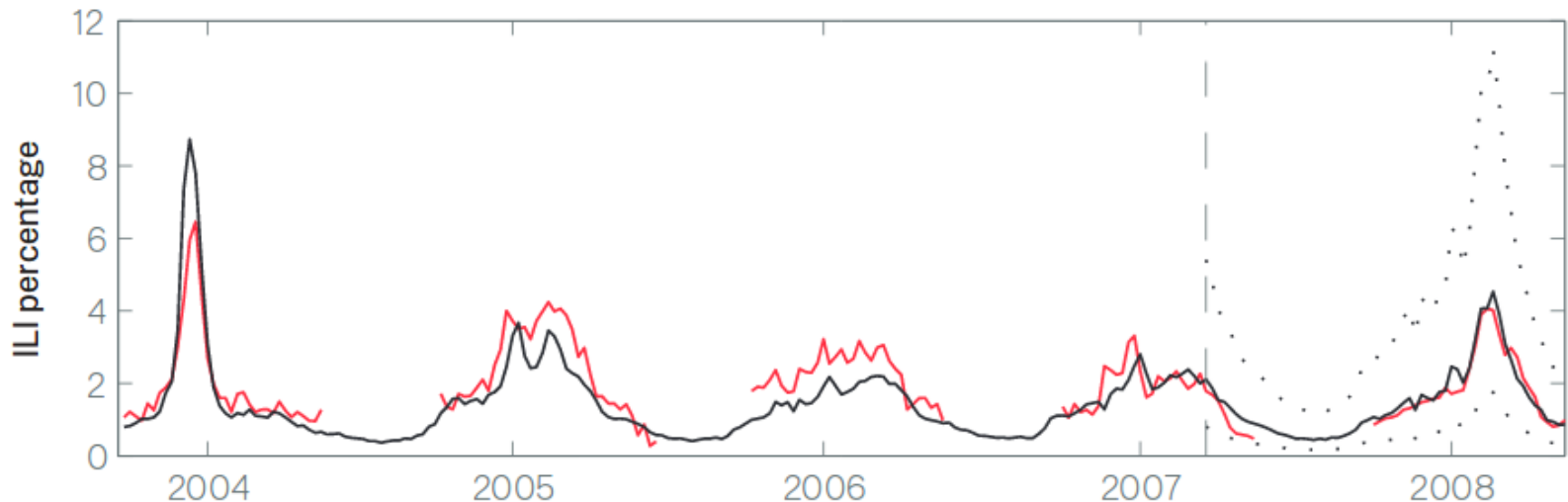
# Extend query model (cont.)

$$CoWeight_Q(w_j^{(d)}) = \ln\left(\frac{1}{\sum_{w_t^{(q)} \in Q} (P(w_j^{(d)} | w_t^{(q)}) + 1)}\right)$$

1. Extract term in query Q
2. Find documents related to any term
3. For each term in each document, use the formula to measure relevance to query Q
4. Using top n highest score term to construct query Q'
5. Search with query Q'

# Application 3: Disease warning

- <https://www.google.org/flutrends>
- Based on related queries
- The number of people looking for information about the disease is proportional to the







25 YEARS ANNIVERSARY  
**SOICT**

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you  
for your  
attentions!



[soict.hust.edu.vn/](http://soict.hust.edu.vn/)



[fb.com/groups/soict](https://fb.com/groups/soict)

