INLS 490-154W: Information Retrieval Systems Design and Implementation

Fall 2009 Web-based course

Retrieval models-1

Relevant or not relevant that is the question

Chirag Shah

School of Information & Library Science (SILS)

UNC Chapel Hill

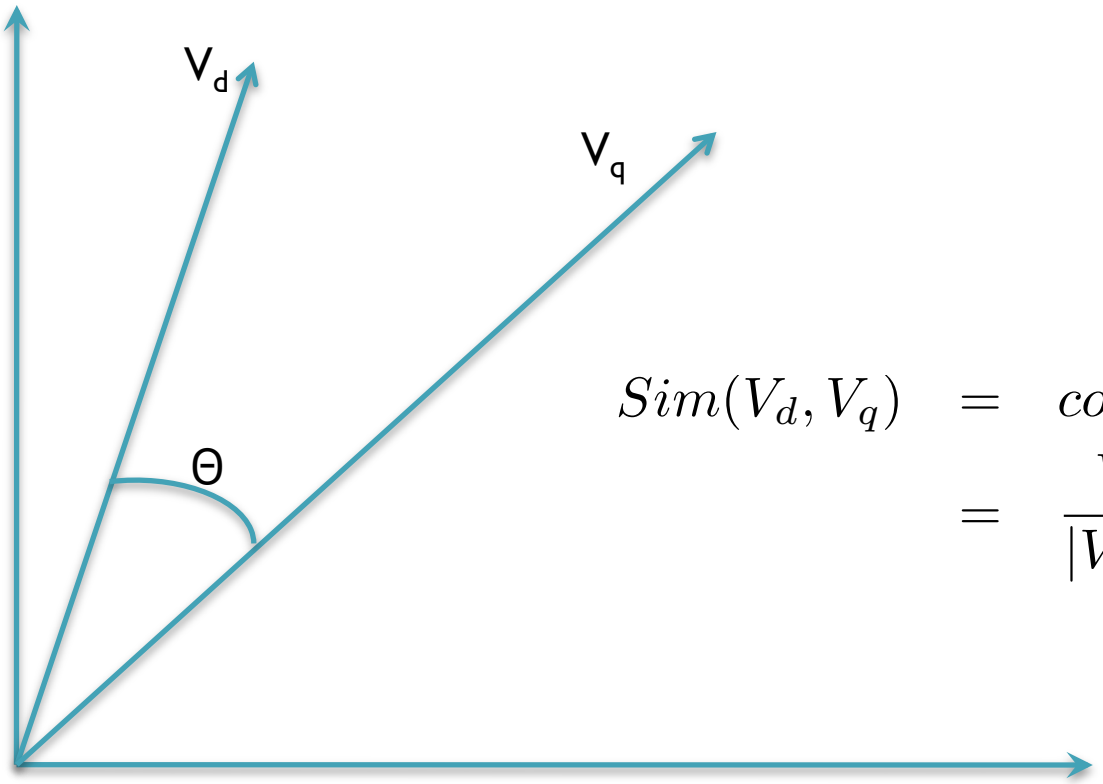
Last time

- IR = matching information need to information
- For Web IR with textual information,
  - Information = text documents
  - Information need = text query
  - Results = set of text documents
- TFIDF is the most common way of weighing terms in IR
- Concepts covered: indexing, query processing, vector space model, TFIDF term weighing, cosine matching

© 2009 Chirag Shah. UNC Chapel Hill

Last time

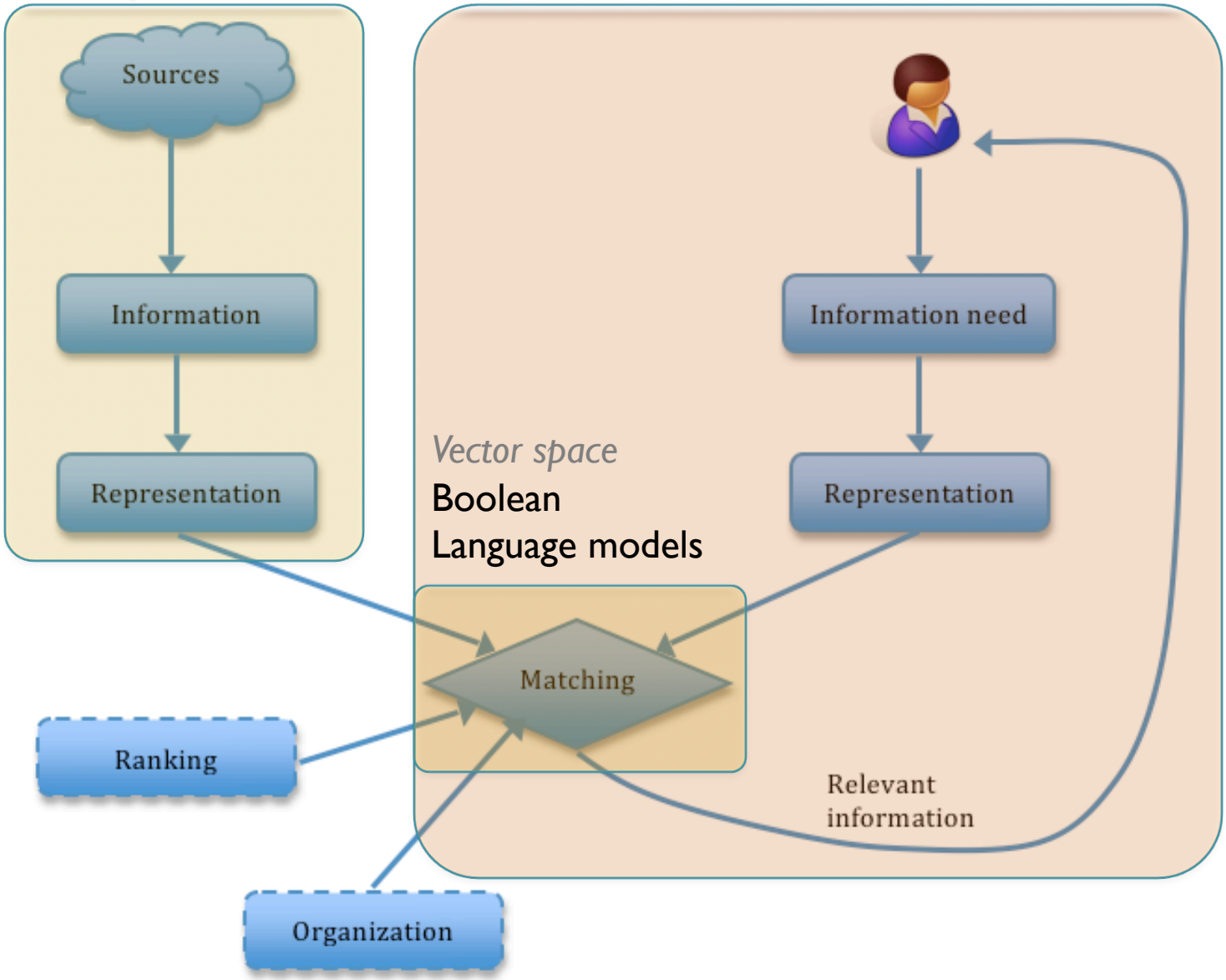
- Vector space model's key idea: a piece of information (document or query) is a vector in high-dimensional space



$$Sim(V_d, V_q) = cosine\theta = \frac{V_d V_q}{|V_d||V_q|}$$

© 2009 Chirag Shah. UNC Chapel Hill

Today's lab



© 2009 Chirag Shah. UNC Chapel Hill

A matter of matching

1. Exact match
   - Query specifies precise retrieval criteria
   - A document matches or fails to match query
   - Result is a set of document without an order (in a true exact match scenario)
2. Best match
   - Query describes good or "best" matching document
   - Every document has some degree of matching
   - Result is a ranked list ordered according to that degree of matching

Source: IR lecture notes
James Allan. UMass Amherst

© 2009 Chirag Shah. UNC Chapel Hill

Boolean retrieval with Lemur

```
#q<id> = <operator>(<query_terms>);
```

- Operators:
  - And Operator: `and(T1 ... Tn)`
  - Boolean And Operator: `band(T1 ... Tn)`
  - Negation Operator: `not(T1)`
  - Boolean And Not Operator: `bandnot (T N)`

© 2009 Chirag Shah. UNC Chapel Hill

Boolean retrieval with Lemur

ParseInQueryOp <param_file> <input_files>

```
<parameters>
        <outputFile>query_parsed.txt</outputFile>
        <docFormat>trec</docFormat>
        <stopwords>stopwords.list</stopwords>
        <stemmer>krovetz</stemmer>
</parameters>
```

StructQueryEval <param_file>

```
<parameters>
        <index>myindex</index>
        <textQuery>query_parsed.txt</textQuery>
        <resultFile>results.txt</resultFile>
        <resultFormat>3col</resultFormat>
        <resultCount>5</resultCount>
        <retModel>inq_struct</retModel>
</parameters>
```


© 2009 Chirag Shah. UNC Chapel Hill

Statistical language models (LM)

- LM describes probability distribution over strings of text
- How likely a given string (observation) in a given language?
  - p_1 = P("a quick brown dog")
  - p_2 = P("dog quick a brown")
  - p_3 = P("un brown dog rapide")

Source: IR lecture notes
James Allan. UMass Amherst

© 2009 Chirag Shah. UNC Chapel Hill

Quick review of probabilities

- P(e) is the probability of event 'e'.
- P(e|C) is the probability of event 'e' happening given condition 'C'.
- $\sum_k P(e) = 1$
- $P(e) + P(\text{not } e) = 1$
- If e_1 and e_2 are independent,

$$P(e_1 \text{ AND } e_2) = P(e_1)P(e_2)$$

$$P(e_1, ..., e_j|C) = \Pi_{i=1}^{j} P(e_i|C)$$

- Bayes' rule

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

© 2009 Chirag Shah. UNC Chapel Hill

Query likelihood model

$$P(q_1 ... q_j | M_D) = \Pi_{i=1}^{k} P(q_i | M_D)$$

What is the probability that a document could have generated the given query?

$$\hat{p}(t|M_d) = \begin{cases} p_{ml}(t,d)^{(1.0-\hat{R}_{t,d})} \times p_{avg}(t)^{\hat{R}_{t,d}} & \text{if } tf_{(t,d)} > 0 \\ \frac{cf_t}{cs} & \text{otherwise} \end{cases}$$

Then,

$$\hat{p}(Q|M_d) = \prod_{t \in Q} \hat{p}(t|M_d) \times \prod_{t \notin Q} 1.0 - \hat{p}(t|M_d)$$

Source: Ponte & Croft, 1998

© 2009 Chirag Shah. UNC Chapel Hill

Language modeling with Lemur

- Kullback-Leibler (KL) divergence
  - Also known as relative entropy
  - Documents are ranked according to the negative of the divergence of the query language model from the document language model
  - $KL(M_Q \mid\mid M_D)$

© 2009 Chirag Shah. UNC Chapel Hill

Language modeling with Lemur

RetEval <param_file>

```
<parameters>
        <index>myindex</index>
        <textQuery>query_parsed.txt</textQuery>
        <resultFile>results.txt</resultFile>
        <resultFormat>3col</resultFormat>
        <resultCount>5</resultCount>
        <retModel>kl</retModel>
</parameters>
```

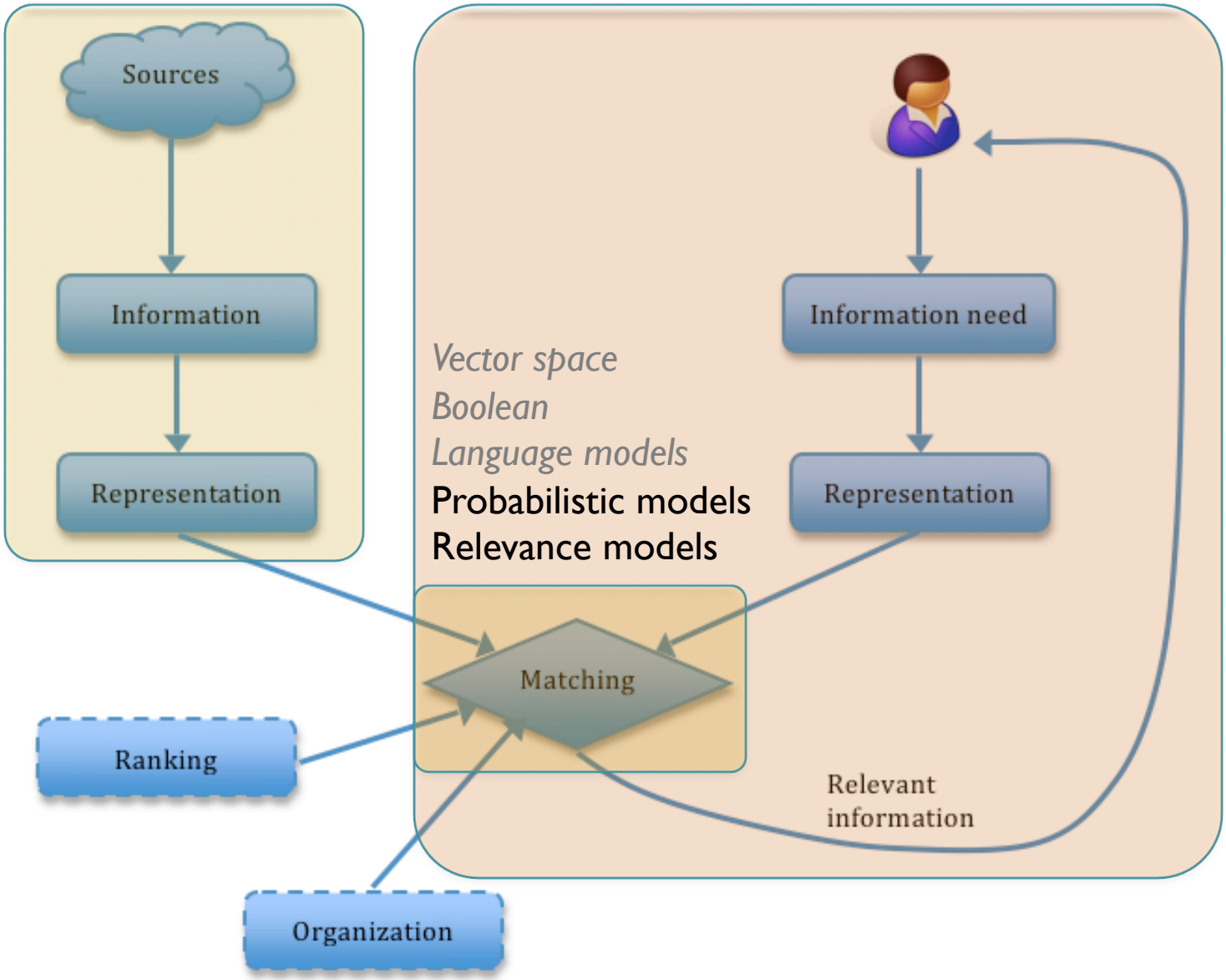

© 2009 Chirag Shah. UNC Chapel Hill

Summary

- Two major ways of matching a query with a collection of documents: exact matching and best (approximate) matching.
- Boolean queries facilitate exact matching.
- Statistical language models are very appealing for best matching scheme.
- Language models estimate the probability that the given query is generated by a document.

© 2009 Chirag Shah. UNC Chapel Hill

Next time



© 2009 Chirag Shah. UNC Chapel Hill