



**INLS 490-154: Information Retrieval Systems
Design & Implementation
Spring 2009**

Structured Query Processing

There's more to a query than meets the eye

February 26, 2009

Chirag Shah

School of Information & Library Science (SILS)

UNC Chapel Hill



Mid-semester review: theoretical concepts

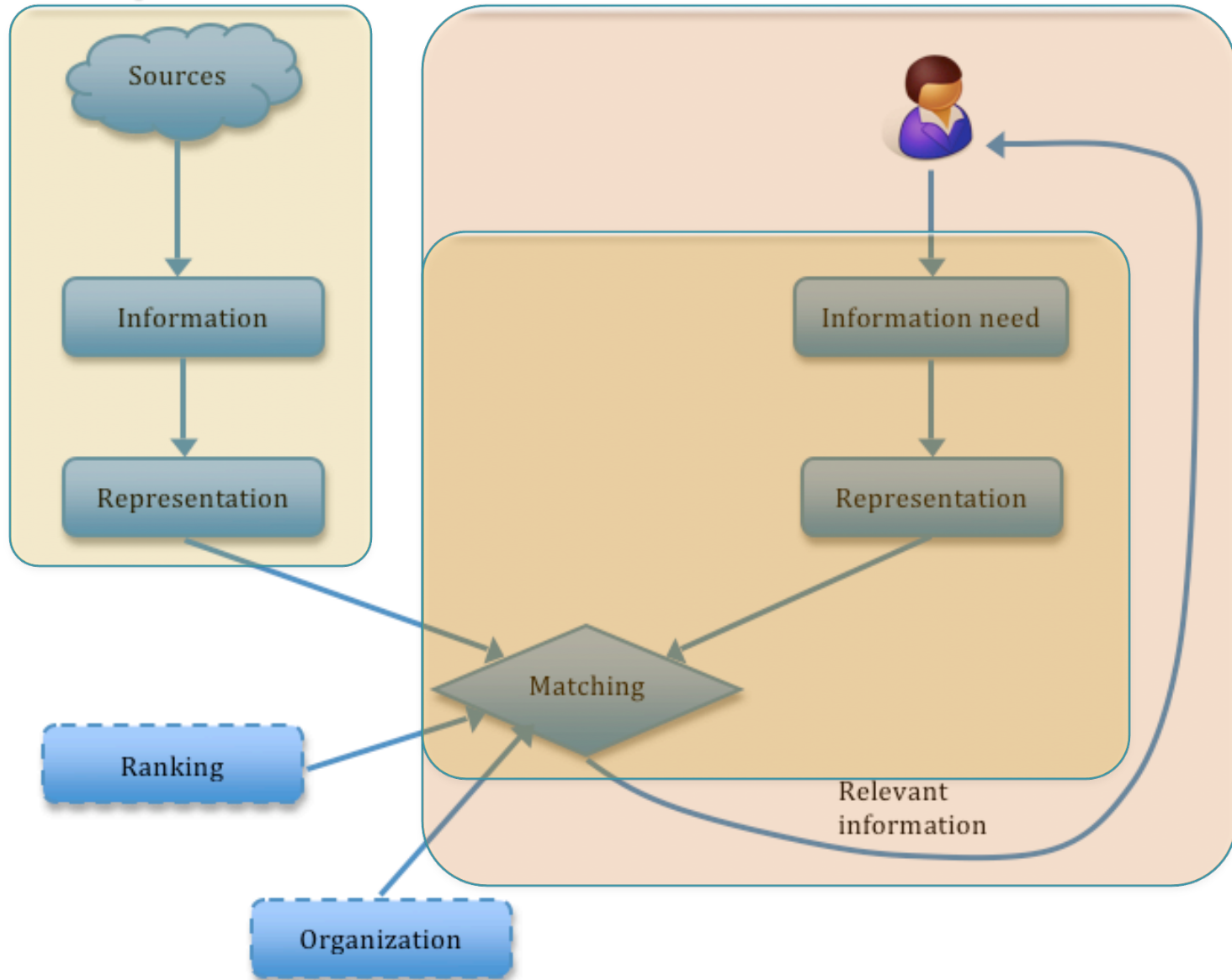
- Structured vs. unstructured information processing
- Serial scanning
- Bag-of-words
- Indexing
- Tokenization
- Stop words removal
- Stemming
- Inverted file structure
- Precision and recall
- Query processing
- Retrieval models: vector space, boolean, language models, probabilistic models, relevance models
- Term weighing: TFIDF, KL, Okapi



Mid-semester review: practical concepts

- MySQL
- PHP access to MySQL database
- Text matching
- Porter and Krovetz stemmers
- ParseToFile
- ParseInQueryOp
- BuildIndex
- dumpindex
- RetEval
- StructQueryEval
- RelFBEval

Today's lab



Structured queries with Lemur

#q1 = #SUM(presidential election Bush Kerry);

#q2 = #WSUM(1 presidential 1 election 1 Bush 1 Kerry);

#q3 = #WSUM(0.5 presidential 0.5 election 0.5 Bush 0.5 Kerry);

#q4 = #WSUM(2 presidential 5 election 0.5 Bush 0.75 Kerry);

Structured queries 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>10</resultCount>
  <retModel>inq_struct</retModel>
</parameters>
```

Retrieval with feedback

RetEval <param_file>

```
<parameters>
  <index>myindex</index>
  <textQuery>query_parsed.txt</textQuery>
  <resultFile>results.txt</resultFile>
  <resultFormat>3col</resultFormat>
  <resultCount>100</resultCount>
  <retModel>k1</retModel>
  <feedbackDocCount>5</feedbackDocCount>
  <feedbackTermCount>10</feedbackTermCount>
</parameters>
```

Extracting term IDs and weights

- Modify RetEval
- Output:
 - <term_id> <term_weight>
- **dumpTerm <index_name> <term_id>**
- Output:
 - <term_name>
 - <doc_id>(<term_count>): <term_positions>



Summary

- Structured queries allow to express user/system to emphasize query terms differently.
- Weights = “importance” of a term?