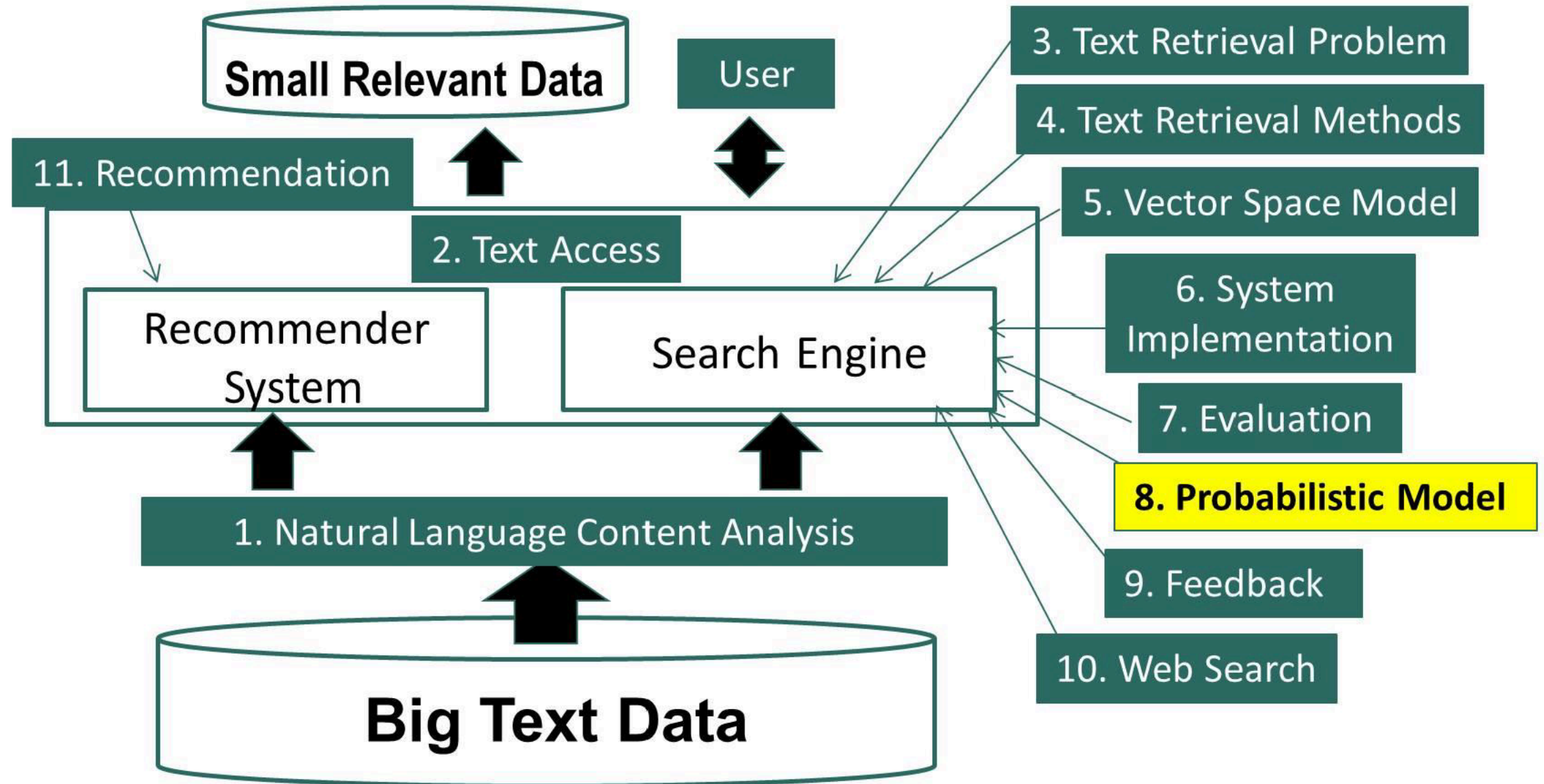


# **Information Retrieval & Text Mining**

## **Probabilistic Retrieval Model: Smoothing Methods**

**Dr. Saeed Ul Hassan**

# Probabilistic Retrieval Model: Smoothing Methods



# Query Likelihood + Smoothing with $p(w | C)$

$$\log p(q | d) = \sum_{\substack{w_i \in d \\ w_i \in q}} c(w, q) \left[ \log \frac{p_{\text{Seen}}(w_i | d)}{\alpha_d p(w_i | C)} \right] + n \log \alpha_d + \boxed{\sum_{i=1}^n \log p(w_i | C)}$$

$$f(q, d) = \sum_{\substack{w_i \in d \\ w_i \in q}} c(w, q) \left[ \log \frac{p_{\text{Seen}}(w_i | d)}{\alpha_d p(w_i | C)} \right] + n \log \alpha_d$$

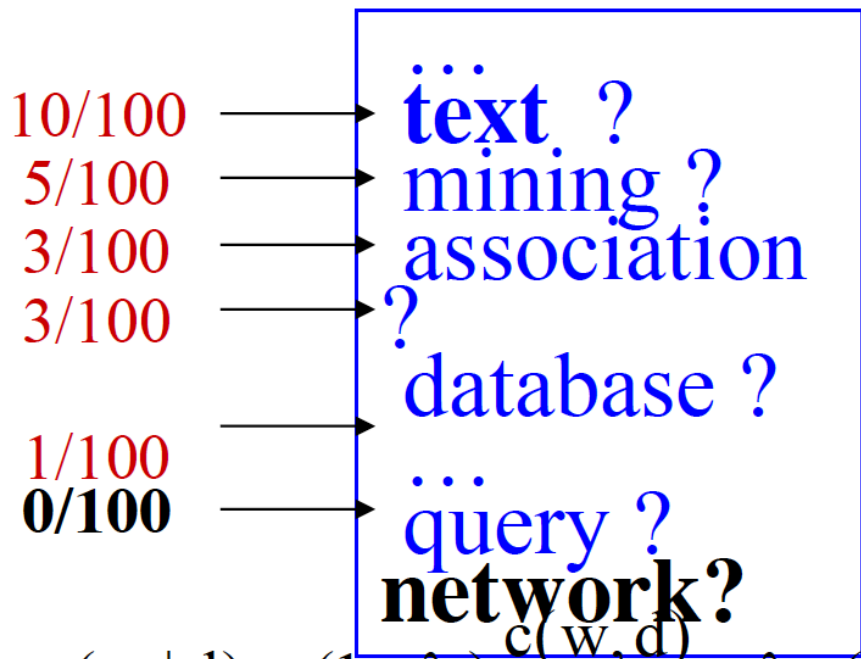
$$\boxed{p_{\text{Seen}}(w_i | d) = ?}$$

$$\boxed{\alpha_d = ?}$$

How to smooth  $p(w | d)$ ?

# Linear Interpolation (Jelinek-Mercer) Smoothing

Unigram LM  $p(w|\theta)=?$



$$p(w | d) = (1 - \lambda) \frac{c(w, d)}{|d|} + \lambda p(w | C)$$

$$p(\text{"text"} | d) = (1 - \lambda) \frac{10}{100} + \lambda * 0.001$$

Document d  
Total #words=100

text 10  
mining 5  
association 3  
database 3  
algorithm 2  
  
query 1  
efficient 1

$$\lambda \in [0,1]$$

$$p(\text{"network"} | d) = \lambda * 0.001$$

Collection LM  
 $P(w|C)$

the 0.1  
a 0.08  
  
computer 0.02  
database 0.01  
  
...  
text 0.001  
network 0.001  
mining 0.0009  
...

# Dirichlet Prior (Bayesian) Smoothing

Unigram LM  $p(w|\theta)=?$

10/100

5/100

3/100

3/100

1/100

0/100

text ?  
mining ?  
association  
?  
database ?  
query ?  
network?

Document d  
Total #words=100

text 10  
mining 5  
association 3  
database 3  
algorithm 2  
query 1  
efficient 1

Collection LM  
 $P(w|C)$

the 0.1  
a 0.08  
computer 0.02  
database 0.01  
text 0.001  
network 0.001  
mining 0.0009  
...

$$p(w|d) = \frac{c(w,d) + \mu p(w|C)}{|d| + \mu} = \frac{|d|}{|d| + \mu} \frac{c(w,d)}{|d|} + \frac{\mu}{|d| + \mu} p(w|C)$$

$$\mu \in [0, +\infty)$$

$$p(\text{"text"}|d) = \frac{10 + \mu * 0.001}{100 + \mu}$$

$$p(\text{"network"}|d) = \frac{\mu}{100 + \mu} * 0.001$$



# Summary

- Two smoothing methods
  - Jelinek-Mercer: Fixed coefficient linear interpolation
  - Dirichlet Prior: Adding pseudo counts; adaptive interpolation
- Both lead to state of the art retrieval functions with assumptions clearly articulated (less heuristic)
  - Also implementing TF-IDF weighting and doc length normalization
  - Has precisely one (smoothing) parameter

# Summary of Query Likelihood Probabilistic Model

- Effective ranking functions obtained using pure probabilistic modeling
  - Assumption 1:  $\text{Relevance}(q,d) = p(R=1 | q,d) \approx p(q | d, R=1) \approx \mathbf{p(q | d)}$
  - Assumption 2: Query words are generated independently
  - Assumption 3: Smoothing with  $p(w | C)$
  - Assumption 4: JM **or** Dirichlet prior smoothing
- Less heuristic compared with VSM
- Many extensions have been made [Zhai 08]

# Additional Readings

- ChengXiang Zhai, *Statistical Language Models for Information Retrieval* (Synthesis Lectures Series on Human Language Technologies), Morgan & Claypool Publishers, 2008.

<http://www.morganclaypool.com/doi/abs/10.2200/S00158ED1V01Y200811HLT001>