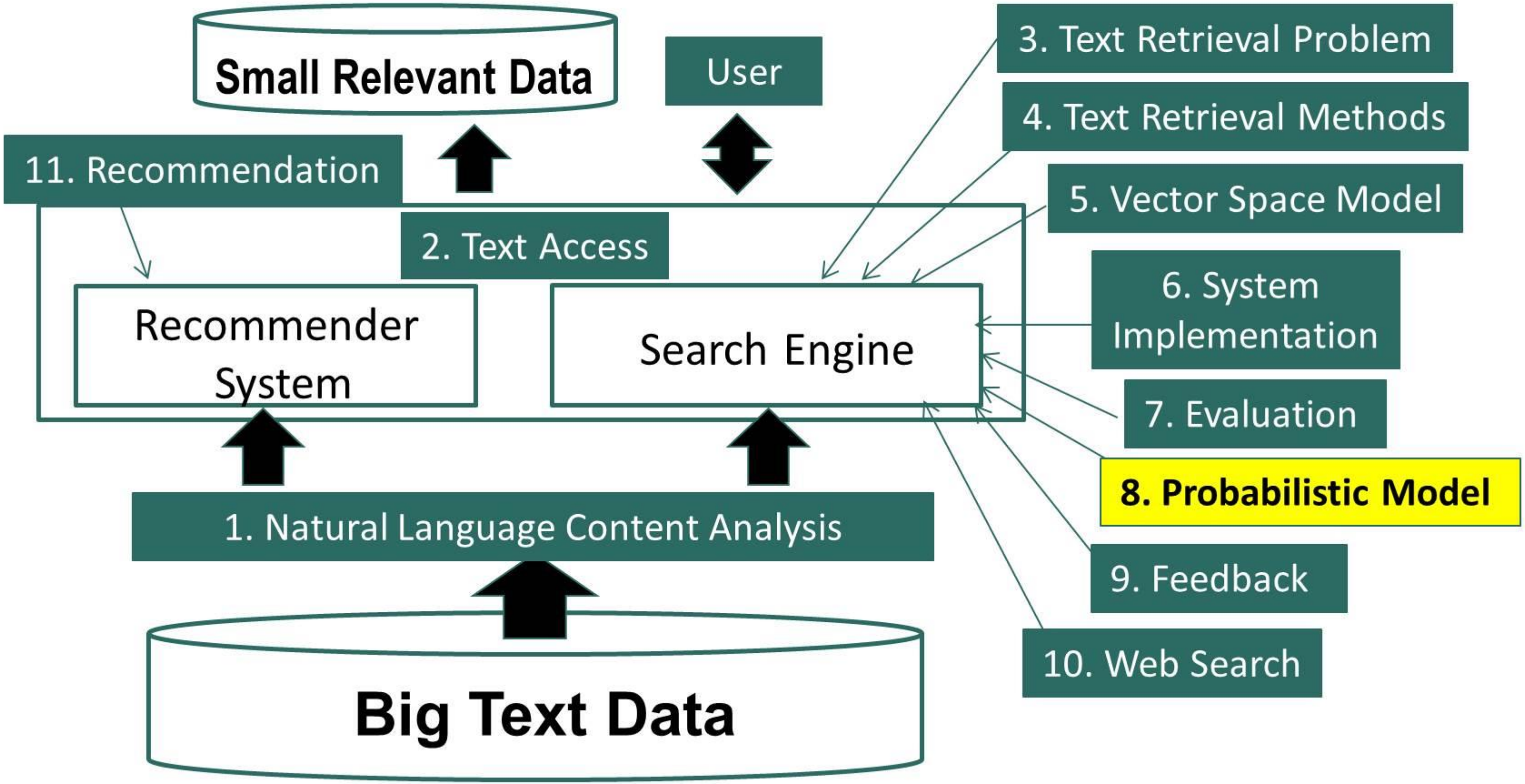
Text Retrieval and Search Engines

Probabilistic Retrieval Model: Smoothing - Part 1 & 2

ChengXiang "Cheng" Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign

Probabilistic Retrieval Model: Smoothing



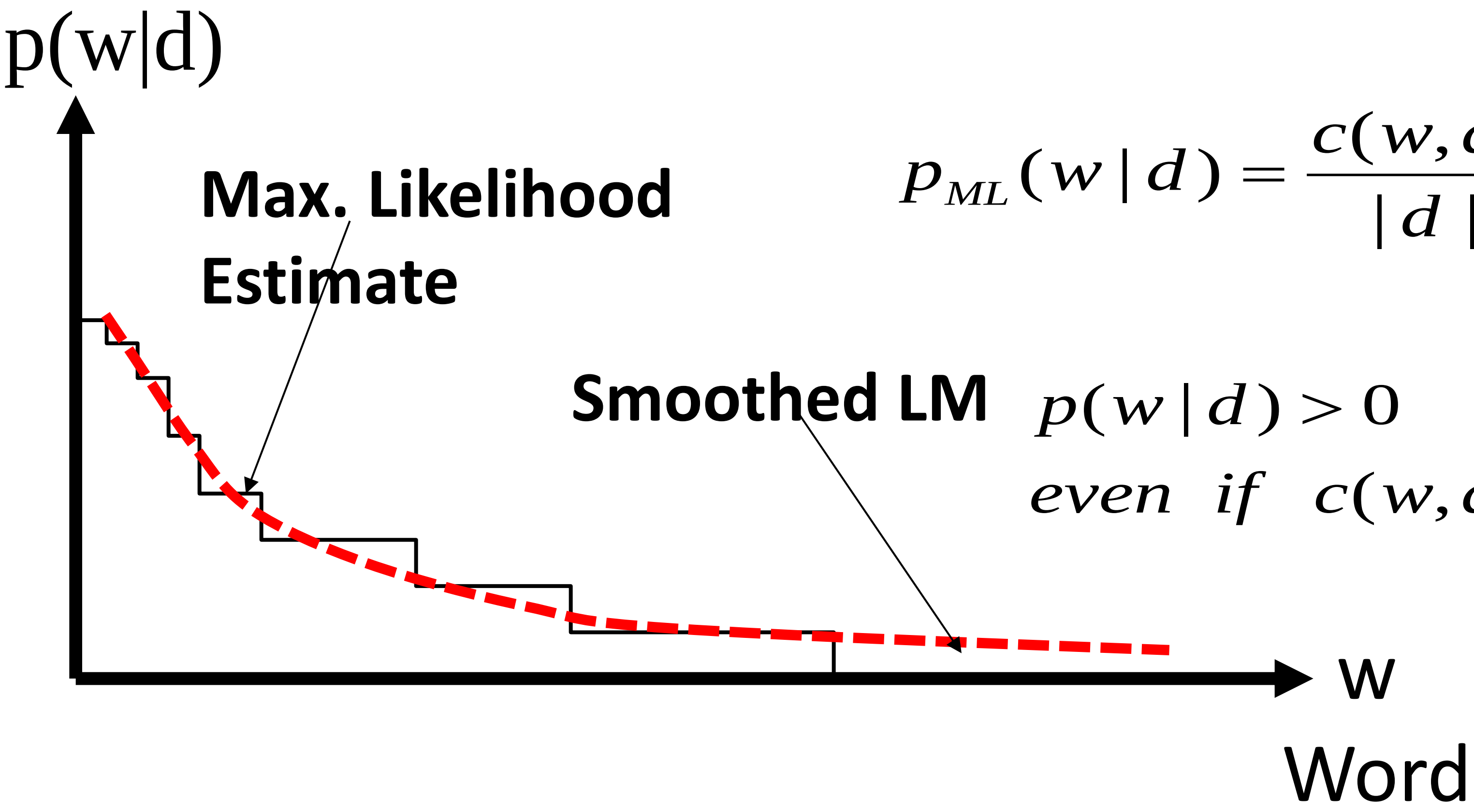
Ranking Function based on Query Likelihood

$$q = w_1 w_2 \ldots w_n \qquad p(q \mid d) = p(w_1 \mid d) \times \ldots \times p(w_n \mid d)$$

$$f(q,d) = \log p(q \mid d) = \sum_{i=1}^{n} \log p(w_i \mid d) = \sum_{w \in V} c(w,q) \log \boxed{p(w \mid d)}$$

How should we estimate *p(w|d)*?

How to Estimate p(w|d)



$$p_{ML}(w \mid d) = \frac{c(w,d)}{|d|}$$

$$p(w \mid d) > 0$$
$$even \ \ if \ \ c(w,d) = 0$$

How to smooth a LM

- Key Question: what probability should be assigned to an unseen word?
- Let the probability of an unseen word be proportional to its probability given by a reference LM
- One possibility: Reference LM = Collection LM

Discounted ML estimate

$$p(w|d) = \begin{cases} p_{Seen}(w|d) & \text{if } w \text{ is seen in } d \\ \alpha_d \, p(w|C) & \text{otherwise} \end{cases}$$

Collection language model

Rewriting the Ranking Function with Smoothing

$$\log p(q\,|\,d) = \sum_{w \in V} c(w,q) \log p(w\,|\,d)$$

$$= \boxed{\sum_{w \in V, \boldsymbol{c(w,d)>0}} c(w,q) \log p_{Seen}(w\,|\,d)} + \boxed{\sum_{w \in V, \boldsymbol{c(w,d)=0}} c(w,q) \log \alpha_d p(w\,|\,C)}$$

Query words **matched** in d

Query words **not matched** in d

$$\sum_{w \in V} c(w,q) \log \alpha_d p(w\,|\,C) - \boxed{\sum_{w \in V, c(w,d)>0} c(w,q) \log \alpha_d p(w\,|\,C)}$$

All query words

Query words **matched** in d

$$= \sum_{w \in V, c(w,d)>0} c(w,q) \log \frac{p_{Seen}(w\,|\,d)}{\alpha_d p(w\,|\,C)} + |q| \log \alpha_d + \sum_{w \in V} c(w,q) \log p(w\,|\,C)$$

Benefit of Rewriting

- Better understanding of the ranking function
  - Smoothing with p(w|C) ➔ TF-IDF weighting + length norm.

TF weighting

Doc length normalization

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

matched query terms

IDF weighting

Ignore for ranking

- Enable efficient computation

Summary

- Smoothing of p(w|d) is necessary for query likelihood
- General idea: smoothing with p(w|C)
  - The probability of an unseen word in d is assumed to be proportional to p(w|C)
  - Leads to a general ranking formula for query likelihood with TF-IDF weighting and document length normalization
  - Scoring is primarily based on sum of weights on matched query terms
- However, how exactly should we smooth?