Language Model

(COSC 488)

Nazli Goharian
nazli@ir.cs.georgetown.edu

See the reference section for the resources used to prepare these lecture notes.

1

Retrieval using Language Model

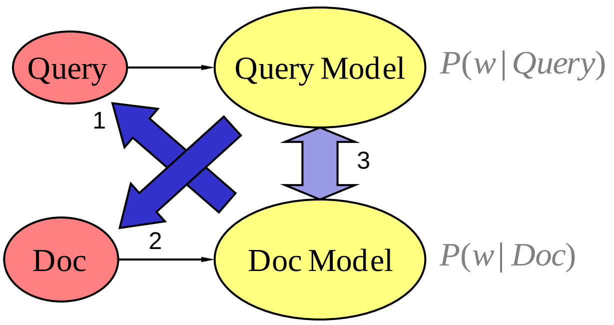
- A probabilistic model of text
  - Documents or queries are modeled based on probability distribution over sequences of words
- *Ponte and Croft's pioneering paper [ACM SIGIR 1998]*
- *Variations studied since then*

- Three main approaches:
  - *Query likelihood model :* generating query from the document language model
  - *Document likelihood model:* generating document from query language model
  - *KL-Divergence model*: Can compare the document and query language models

2

Retrieval Using Language Models

(from: C. Manning, P. Raghavan & H. Schütze, Introduction to Information Retrieval, Cambridge University Press., 2008)



Retrieval: Query likelihood (1), Document likelihood (2), Model comparison (3)

Query Likelihood Scoring Method:

Computing $p(Q|D)$ or $p(Q|\theta D)$

- Goal: determine which document or document model best derives (specific) query Q
- A query is sample of words drawn from a document based on the model defined for the document (document language model θD)
- Documents are then ranked based on their likelihood of giving (generating) that query
- Document models that give a higher probability to the query indicate having more terms of the query (capturing the notion of *TF*) score$(Q,D) = p(Q|\theta D)$

Query Likelihood Model

- Unigram query likelihood

$$P(Q \mid \theta D) = \prod_{i=1}^{n} P(q_i \mid D)$$

$$P(q_i \mid D) = \frac{tf_{q_i,D}}{|D|}$$

⟸ *Maximum liklihood (ML) estimate, defined as:* Tf of query term appearing in doc divided by document length

Example:

Q: "computer virus,"

p(computer|D) = 0.1, p(virus|D) = 0.05 ➔ *p(Q|θD) = 0.1 ∗ 0.05 = 0.005*

- Problems:
  - Results in zero if a term is missing in document (estimation problem)
  - Document may be relevant to query but the query term is absent from document (data sparsity problem)

Query: Virus
Topic of document D1: Epidemic
Assume P(virus|D1) = 0

➔ **Need smoothing!**

5

Need for Smoothing: Example

(example from: Grossman & Frieder, Information Retrieval Algorithms and Heuristics, 1998, 2nd Edition, Springer, 2004.)

- If a term in a query does not occur in a document, the whole similarity measure becomes *zero*
- Example:
  - Q: "gold silver truck"
  - D1: "Shipment of gold damaged in a fire"
  - D2: "Delivery of silver arrived in a silver truck"
  - D3: "Shipment of gold arrived in a truck"
- Term *Silver* does not appear in D_1. Similarly, *silver* does not appear in D_3 and *gold* does not appear in D_2.
- This would result in Score=0 for all 3 documents.

$$p_{ml}(silver \mid \theta D_i) = \frac{tf(silver, D_i)}{|D_i|} = 0$$

Need for Smoothing: Example

(example from: Viktor Lavrenko and Chengxiang Zhai)

Query =	"the	algorithms	for	data	mining"
d1:	**0.04**	**0.001**	**0.02**	**0.002**	**0.003**
d2:	**0.02**	**0.001**	**0.01**	**0.003**	**0.004**

p("algorithms"|d1) = p("algorithm"|d2)
p("data"|d1) < p("data"|d2)
p("mining"|d1) < p("mining"|d2)

But p(q|d1)>p(q|d2)!

We should make p("the") and p("for") less different for all docs.

Variations of Language Modeling Approach

- Variations of basic language modeling approach, based on:
  - Estimating document model *θD*
  - Various smoothing methods *(Jelinek-Mercer, Dirichlet, ...)*
  - Document Prior P(D) *(document features such as page rank, url length, time, anchor text....)*

Smoothing Query Likelihood Model

- To deal with the *estimation* problem and *data sparsity*, smooth the probability estimates by:
  - Lowering the probability estimate of the terms in document
  - Assigning probabilities to unseen terms in document
    (calculated generally based on the entire collection – collection language model / background language / background probability)

$$P(q_i \mid D) = (1-\alpha_D)P(q_i \mid D) + \alpha_D P(q_i \mid C)$$

$$P(q_i \mid D) = (1-\alpha_D)\frac{tf_{q_i,D}}{|D|} + \alpha_D \frac{tf_{q_i,C}}{|C|}$$

occurrences of query term i in the collection

No. of terms in the entire collection

- Various smoothing based on how to handle α_D

9

Jelinek-Mercer Smoothing

- Set the coefficient to a constant $\alpha_D = \lambda \in [0,1]$

$$P(q_i \mid D) = (1-\lambda)\frac{tf_{q_i,D}}{|D|} + \lambda \frac{tf_{q_i,C}}{|C|}$$

$\lambda = 0$ → Query similar to Boolean AND

Larger λ → Query similar to Boolean OR

In TREC evaluations: λ = 0.1 for short queries

λ = 0.7 for long queries

(if no training data, generally: 0.5)

10

Jelinek-Mercer Smoothing (Cont'd)

$$P(q_i \mid D) = (1-\lambda)\frac{tf_{q_i,D}}{|D|} + \lambda \frac{tf_{q_i,C}}{|C|}$$

$$P(Q \mid D) = \prod_{i=1}^{n}((1-\lambda)\frac{tf_{q_i,D}}{|D|} + \lambda \frac{tf_{q_i,C}}{|C|})$$

$$\log P(Q \mid D) = \sum_{i=1}^{n}\log\,((1-\lambda)\frac{tf_{q_i,D}}{|D|} + \lambda \frac{tf_{q_i,C}}{|C|})$$

Dirichlet Smoothing

- Considers document length (μ terms are added to increase the chance of match)

$$P(q_i \mid D) = \frac{tf_{q_i,D} + \mu \frac{tf_{q_i,C}}{|C|}}{|D| + \mu}$$

$$\log P(Q \mid D) = \sum_{i=1}^{n}\log \frac{tf_{q_i,D} + \mu \frac{tf_{q_i,C}}{|C|}}{|D| + \mu}$$

- Longer documents are impacted less by μ (should be tuned or pick *average document length*)

- Comparable to well-tuned retrieval models of *TF-IDF* with pivoted length normalization, and *BM25*

KL-Divergence Model:

Computing $P(\theta Q||\theta D)$

- A state-of-the-art LM approach to rank documents
- Similar concept as to *vector space model* ; however, probabilistic representation of text and distance function
- The difference between *document model* and *query model* *(relevance model)* is measured

$$score(D,Q) = \sum_{m \in v} P(w \mid \theta Q) \log p(w \mid \theta D)$$

$$score(D,Q) = \sum_{m \in v} \underbrace{\frac{f_{w,Q}}{|Q|}}_{\text{ML Estimate}} \log \underbrace{p(w \mid \theta D)}_{\text{Use Dirichlet smoothing}}$$

Language Models vs. Traditional Retrieval Models

- Query likelihood with Dirichlet smoothing offers similar performance to TF-IDF & BM25 retrieval functions
- Sophisticated language models can be computationally expensive

References

- Ponte and Croft's pioneering paper [ACM SIGIR 1998]
- D. Grossman & O. Frieder, Information Retrieval Algorithms and Heuristics, 1998, 2nd Edition, Springer, 2004.
- ChengXiang Zhai, Statistical Language Models for Information Retrieval: A Critical Review, Foundation & Trends in Information Retrieval, 2008
- C. Manning, P. Raghavan & H. Schütze, Introduction to Information Retrieval, Cambridge University Press., 2008.
- W. Croft, D. Metzler, T. Strohman, Search Engines: Information Retrieval in Practice, Addison Wesley, 2010
- S. Buttcher, C. Clarke, G. Cormack, Information Retrieval: Implementing and Evaluating search Engines, Addison Wesley, 2010