Information Retrieval

Weike Pan

The slides are **adapted from those provided by Prof. Hinrich Schütze** at University of Munich (http://www.cis.lmu.de/~hs/teach/14s/ir/).

Chapter 12 Language models for information retrieval

- 12.1 Language models
- 12.2 The query likelihood model
- 12.3 Language modeling versus other approaches in IR
- 12.4 Extended language modeling approaches
- 12.5 References and further reading

Outline

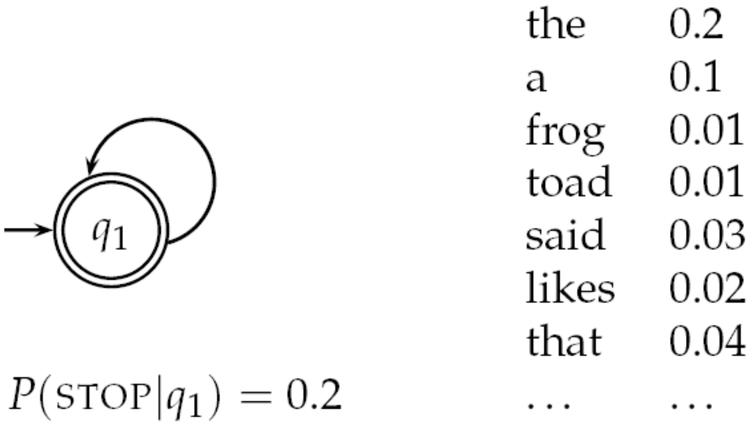
- 12.1 Language models
- 12.2 The query likelihood model
- 12.3 Language modeling versus other approaches in IR
- 12.4 Extended language modeling approaches
- 12.5 References and further reading

12.1 Language models

- In language model (LM), we view the document as a generative model that generates the query.

- Steps:
  - Define the precise generative model we want to use
  - Estimate parameters (different parameters for each document's model)
  - Smooth to avoid zeros
  - Apply to query and find document most likely to have generated the query
  - Present most likely document(s) to user

12.1 Language models



the	0.2
a	0.1
frog	0.01
toad	0.01
said	0.03
likes	0.02
that	0.04
…	…

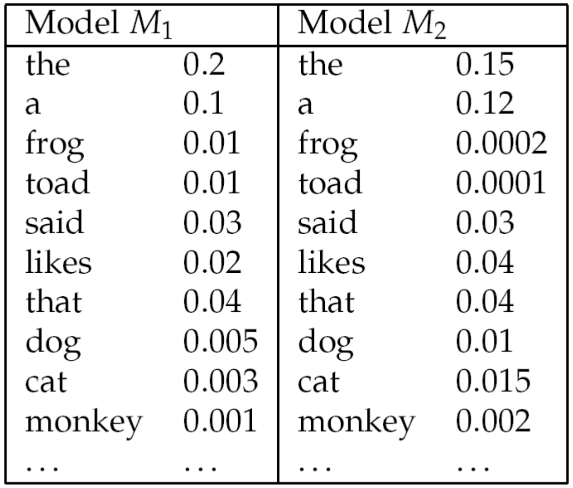
$P(\text{STOP}|q_1) = 0.2$

► **Figure 12.2** A one-state finite automaton that acts as a unigram language model. We show a partial specification of the state emission probabilities.

Example 12.1: To find the probability of a word sequence, we just multiply the probabilities which the model gives to each word in the sequence, together with the probability of continuing or stopping after producing each word. For example,

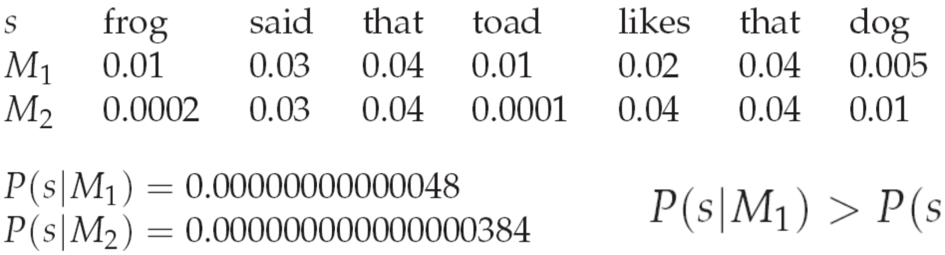
$$\begin{aligned} P(\text{frog said that toad likes frog}) &= (0.01 \times 0.03 \times 0.04 \times 0.01 \times 0.02 \times 0.01) \\ &\quad \times (0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.2) \\ &\approx 0.000000000001573 \end{aligned}$$

12.1 Language models



Model M_1		Model M_2	
the	0.2	the	0.15
a	0.1	a	0.12
frog	0.01	frog	0.0002
toad	0.01	toad	0.0001
said	0.03	said	0.03
likes	0.02	likes	0.04
that	0.04	that	0.04
dog	0.005	dog	0.01
cat	0.003	cat	0.015
monkey	0.001	monkey	0.002
…	…	…	…

► **Figure 12.3** Partial specification of two unigram language models.



s	frog	said	that	toad	likes	that	dog
M_1	0.01	0.03	0.04	0.01	0.02	0.04	0.005
M_2	0.0002	0.03	0.04	0.0001	0.04	0.04	0.01

$$P(s|M_1) = 0.00000000000048$$
$$P(s|M_2) = 0.000000000000000384$$

$$P(s|M_1) > P(s|M_2)$$

document d1 is “more relevant” to the query

Outline

- 12.1 Language models
- 12.2 The query likelihood model
- 12.3 Language modeling versus other approaches in IR
- 12.4 Extended language modeling approaches
- 12.5 References and further reading

12.2 The query likelihood model

- Each document is treated as (the basis for) a language model.
- Given a query *q*, rank documents based on *P(d|q)*

$$P(d|q) = \frac{P(q|d)P(d)}{P(q)}$$

- *P(q)* is the same for all documents
- *P(d)* is the prior -- often treated as the same for all *d,* i.e., uniform prior
  - But we can give a higher prior to “high-quality” documents, e.g., those with high PageRank.
- *P(q|d)* is the probability of *q* given *d*.

12.2 The query likelihood model

- In the LM approach to IR, we attempt to model the query generation process.
- That is, we rank documents according to $P(q|d)$
- How do we compute $P(q|d)$?

12.2 The query likelihood model

- Conditional independence assumption:

$$P(q|M_d) = P(\langle t_1, \ldots, t_{|q|} \rangle | M_d) = \prod_{1 \leq k \leq |q|} P(t_k|M_d)$$

- This is equivalent to:

$$P(q|M_d) = \prod_{\text{distinct term } t \text{ in } q} P(t|M_d)^{\mathrm{tf}_{t,q}}$$

↘ term frequency of term *t* in *q*

- Parameter estimation:

$$\hat{P}(t|M_d) = \frac{\mathrm{tf}_{t,d}}{|d|}$$

→ term frequency of term *t* in document *d*

→ length of document *d*

12.2 The query likelihood model

- For a document that does not contain a certain term *t* in a query *q*, then the probability will be **zero**.

- **Jelinek-Mercer smoothing**

the number of occurrences of term *t* in the collection

$$\lambda P(t|M_d) + (1-\lambda)P(t|M_c)$$

$$\hat{P}(t|M_c) = \frac{\mathrm{cf}_t}{T}$$

the total number of tokens in the collection

- **Dirichlet smoothing**

term frequency of *t* in *d* →

$$\frac{\mathrm{tf}_{t,d} + \alpha\hat{P}(t|M_c)}{L_d + \alpha}$$

length of document *d*

12.2 The query likelihood model

- Jelinek-Mercer smoothing or Dirichlet smoothing?
  - Dirichlet performs better for keyword queries
  - Jelinek-Mercer performs better for verbose (冗长的) queries

  - Both models are sensitive to the smoothing parameters -- you shouldn’t use these models without parameter tuning

12.2 The query likelihood model

- **Example** (Jelinek-Mercer smoothing)
  - Collection: d1 and d2
  - d1: Jackson was one of the most talented entertainers of all time
  - d2: Michael Jackson anointed himself King of Pop
  - q: Michael Jackson

$$\lambda = 1/2$$

$$P(q|d_1) = [(0/11 + 1/18)/2] \cdot [(1/11 + 2/18)/2] \approx 0.003$$

$$P(q|d_2) = [(1/7 + 1/18)/2] \cdot [(1/7 + 2/18)/2] \approx 0.013$$

Ranking: $d_2 > d_1$

12.2 The query likelihood model

- **Exercise** (Jelinek-Mercer smoothing)
  - Collection: d1 and d2
  - d1: Xerox reports a profit but revenue is down
  - d2: Lucene narrows quarter loss but revenue decreases further
  - q: revenue down

Outline

- 12.1 Language models
- 12.2 The query likelihood model
- 12.3 Language modeling versus other approaches in IR
- 12.4 Extended language modeling approaches
- 12.5 References and further reading

12.3 Language modeling versus other approaches in IR

- BM25/LM: based on probability theory
- Vector space: based on similarity, a geometric/linear algebra notion

- Term frequency is directly used in all the three models
- Length normalization

- IDF: BM25/vector space uses it directly
- LMs: Mixing term and collection frequencies has an effect similar to IDF. Terms rare in the general collection, but common in some documents will have a greater influence on the ranking.

- Collection frequency (LMs) vs. document frequency (BM25, vector space)

Summary

- 12.1 Language models
- 12.2 The query likelihood model
- 12.3 Language modeling versus other approaches in IR
- 12.4 Extended language modeling approaches
- 12.5 References and further reading