Information Retrieval

INFO 4300 / CS 4300

- Retrieval models
  - Older models
    - Boolean retrieval
    - Vector Space model
  - Probabilistic Models
    - BM25
    - **Language models**
  - Combining evidence
    - Inference networks
    - Learning to Rank

Language Model

- *Unigram language model*
  - probability distribution over the words in a language
  - generation of text consists of pulling words out of a “bucket” according to the probability distribution and replacing them
- *N-gram language model*
  - some applications use bigram and trigram language models where probabilities depend on previous words

Language Model

- A *topic* in a document or query can be represented as a language model
  - i.e., words that tend to occur often when discussing a topic will have high probabilities in the corresponding language model
- *Multinomial* distribution over words
  - text is modeled as a finite sequence of words, where there are *t* possible words at each point in the sequence
  - commonly used, but not only possibility
  - doesn’t model *burstiness*

LMs for Retrieval

- 3 possibilities:
  - probability of generating the query text from a document language model
  - probability of generating the document text from a query language model
  - comparing the language models representing the query and document topics
- Models of topical relevance

Query-Likelihood Model

- Rank documents by the probability that the query could be generated by the document model (i.e. same topic)
- Start with a query, so calculate P(D|Q) to rank the documents
- Use Bayes' Rule

$$p(D|Q) \stackrel{rank}{=} P(Q|D)P(D)$$

- Assuming prior is uniform, unigram model

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

Estimating Probabilities

- Obvious estimate for unigram probabilities is

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

- *Maximum likelihood estimate*
  - makes the observed value of $f_{q_i,D}$ most likely
- Problem: If query words are missing from document, score for the document will be 0
  - Missing 1 out of 6 query words (score of 0) is the same as missing 5 out of 6

Smoothing

- Document texts are a *sample* from the language model
  - Missing words should not have zero probability of occurring
- *Smoothing* is a technique for estimating probabilities for missing (or unseen) words
  - lower (or *discount*) the probability estimates for words that are seen in the document text
  - assign that "left-over" probability to the estimates for the words that are not seen in the text

Estimating Probabilities

- Estimate for unseen words is $\alpha_D P(q_i|C)$
  - $P(q_i|C)$ is the probability for query word i in the *collection* language model for collection C (background probability)
  - α_D is a parameter
- Estimate for words that occur in a query is

$$(1 - \alpha_D)\, P(q_i|D) + \alpha_D\, P(q_i|C)$$

- Different forms of estimation come from different α_D

Jelinek-Mercer Smoothing

- α_D is a constant, λ
- Gives estimate of

$$p(q_i|D) = (1-\lambda)\frac{f_{q_i,D}}{|D|} + \lambda\frac{c_{q_i}}{|C|}$$

- Ranking score

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

- Use logs for convenience
  - accuracy problems multiplying small numbers

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

Where is *tf.idf* Weight?

$$\begin{aligned}
\log P(Q|D) &= \sum_{i=1}^{n}\log((1-\lambda)\frac{f_{q_i,D}}{|D|} + \lambda\frac{c_{q_i}}{|C|}) \\
&= \sum_{i:f_{q_i,D}>0}\log((1-\lambda)\frac{f_{q_i,D}}{|D|} + \lambda\frac{c_{q_i}}{|C|}) + \sum_{i:f_{q_i,D}=0}\log(\lambda\frac{c_{q_i}}{|C|}) \\
&= \sum_{i:f_{q_i,D}>0}\log\frac{((1-\lambda)\frac{f_{q_i,D}}{|D|} + \lambda\frac{c_{q_i}}{|C|})}{\lambda\frac{c_{q_i}}{|C|}} + \sum_{i=1}^{n}\log(\lambda\frac{c_{q_i}}{|C|}) \\
&\stackrel{rank}{=} \sum_{i:f_{q_i,D}>0}\log\left(\frac{((1-\lambda)\frac{f_{q_i,D}}{|D|}}{\lambda\frac{c_{q_i}}{|C|}} + 1\right)
\end{aligned}$$

- proportional to the term frequency, inversely proportional to the collection frequency

Dirichlet Smoothing

- α_D depends on document length

$$\alpha_D = \frac{\mu}{|D|+\mu}$$

- Gives probability estimation of

$$p(q_i|D) = \frac{f_{q_i,D} + \mu\frac{c_{q_i}}{|C|}}{|D|+\mu}$$

- and document score

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

Query Likelihood Example

- For the term "president"
  - $f_{qi,D} = 15$, $c_{qi} = 160{,}000$
- For the term "lincoln"
  - $f_{qi,D} = 25$, $c_{qi} = 2{,}400$
- number of word occurrences in the document |d| is assumed to be 1,800
- number of word occurrences in the collection is 10^9
  - 500,000 documents times an average of 2,000 words
- μ = 2,000

Query Likelihood Example

$$
\begin{aligned}
QL(Q,D) &= \log \frac{15 + 2000 \times (1.6 \times 10^5/10^9)}{1800 + 2000} \\
&\quad + \log \frac{25 + 2000 \times (2400/10^9)}{1800 + 2000} \\
&= \log(15.32/3800) + \log(25.005/3800) \\
&= -5.51 + -5.02 = -10.53
\end{aligned}
$$

- Negative number because summing logs of small numbers

Query Likelihood Example

Frequency of "president"	Frequency of "lincoln"	QL score	
15	25	-10.53	1
15	1	-13.75	3
15	0	-19.05	5
1	25	-12.99	2
0	25	-14.40	4

BM25 comparison

- Effect of term frequencies

Frequency of "president"	Frequency of "lincoln"	BM25 score		QL
15	25	20.66	1	1
15	1	12.74	**4**	3
15	0	5.00	5	5
1	25	18.2	2	2
0	25	15.66	**3**	4