Retrieval Models

CISC489/689-010, Lecture #9

Wednesday, March 11th

Ben Carterette

Retrieval Models Review

- We've discussed four models so far:
  - Boolean
  - Vector space
  - Binary Independence Model
  - 2-Poisson and its approximation BM25
- Today:
  - Language models

Language Models

- A *language model* is a probability distribution over strings of text.
  - p_1 = P("language model")
  - p_2 = P("probability distribution")
  - p_3 = P("a language model is a probability distribution")
  - $p_2 > p_1 > p_3$?
- Every possible string has some probability.
- Probabilities add to 1.

Unigram Language Models

- A *unigram language model* is a probability distribution over words.
  - P("language"), P("model")
  - P("probability"), P("distribution")
  - P("a"), P("is")
- Each word has a probability, probabilities sum to 1.
- Assumption: words are independent, order doesn't matter.
  - P("language model") = P("language")P("model") = P("model language")
  - P("a language model is a probability distribution") = P("a probability distribution is a language model")
  - "bag of words" model.

Bigram Language Models

- A *bigram language model* is a probability distribution over pairs of words.
  - P("language model")
  - P("probability distribution")
- Or a collection of unigram language models, each conditional on a word.
  - P("model" | "language")
  - P("distribution" | "probability")
- P("a language model is a probability distribution") = P("a")P("language" | "a")P("model" | "language")…P("distribution" | "probability")

Using Language Models

- We have a language sample *M* that we want to model.
  - Could be a collection of articles, a single article, transcripts of conversations, etc.
- *M* "generates" strings of text *s* with some probability P(s | M).
  - If we have a unigram model and s = "w_1 w_2 …", then
    - P(s | M) = P(w_1 | M)P(w_2 | M)…
  - If a bigram model,
    - P(s | M) = P(w_1 | M)P(w_2 | M, w_1)P(w_3 | M, w_2)…

Language Model Example

The majority of Americans consider tobacco advertising a major influence in promoting the killer habit. Approximately 57% of the public thinks that cigarette advertising causes people to smoke. Also, 47% thinks that cigarette advertising makes it harder for smokers to give up the habit. If the tobacco industry didn't agree with these stats it wouldn't concentrate so heavily on using young models in its ads.

Unigram model

makes harder smoke cigarette advertising that wouldn't 47% 57% makes

if on the harder the heavily that of 57% of

Bigram model

that cigarette advertising causes people to give up the tobacco

tobacco industry didn't agree with these stats it wouldn't concentrate

Language Models for IR

- Each document D is a sample of language.
  - Consider all possible strings the author could have written while producing the document.
  - Some strings are more likely than others
    - Depending on topic, writing style, reading level, etc.
  - P(s | D) is the probability that the author would write string s.
- Given a query Q and document D,
  - What is the probability that the author of D would have written Q?: P(Q | D)
  - What is the probability that the user submitting Q would have written D?: P(D | Q)

Language Model Decisions

- Type of model:
  - Unigram, bigram, more complex model?
  - Multinomial, multiple Bernoulli?
- Estimating model parameters:
  - Maximum likelihood, smoothing.
  - Translation models, parsing, background models.
- Using the model for ranking:
  - Query-likelihood, document-likelihood, divergence of query and document models.

Query-Likelihood Model

- Rank documents by the probability that the query could be generated by the document model (i.e. same topic)
- Given query, start with P(D|Q)
- Using Bayes' Rule

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

- Assuming prior is uniform, unigram model

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

Unigram LM Query-Likelihood

The majority of Americans consider tobacco advertising a major influence in promoting the killer habit. Approximately 57% of the public thinks that cigarette advertising causes people to smoke. Also, 47% thinks that cigarette advertising makes it harder for smokers to give up the habit. If the tobacco industry didn't agree with these stats it wouldn't concentrate so heavily on using young models in its ads.

Query

tobacco advertising

P(t | D) = 2/65 3/65 P(Q | D) = 6/4225

tobacco companies

P(t | D) = 2/65 0/65 P(Q | D) = 0

tobacco companies and the young

P(Q | D) = 0

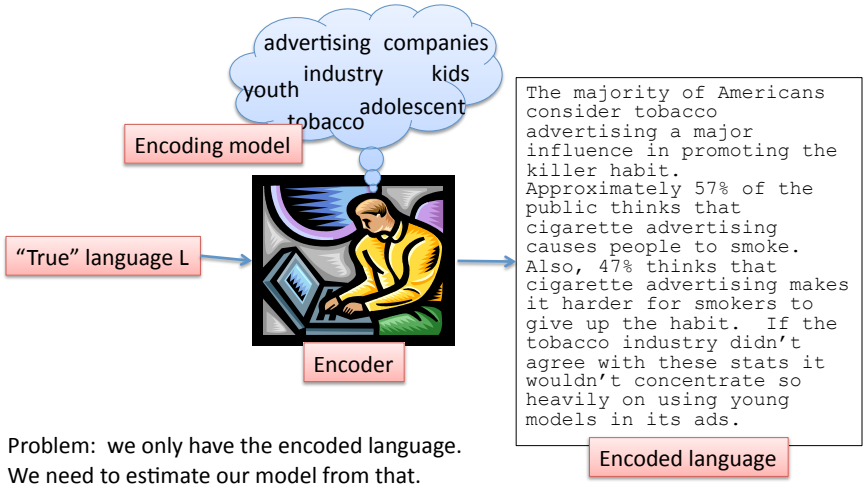
Estimating Probabilities

- 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
- If query words are missing from document, score will be zero
  - Missing 1 out of 4 query words same as missing 3 out of 4

Language Generation as Encoding



Problem: we only have the encoded language.
We need to estimate our model from that.

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
- Gives every term some probability of occurring in every document.

Simple Smoothing Methods

- Laplace correction:
  - Add 1 to every term count: $P(w \mid D) = \frac{tf_{w,D} + 1}{|D| + |V|}$
  - Gives too much weight to unseen terms.
- Lindstone correction:
  - Add small number ε to every term count:

$$P(w \mid D) = \frac{tf_{w,D} + \varepsilon}{|D| + \varepsilon |V|}$$

- Absolute discounting:
  - Subtract small number ε from every term count, renormalize.

Smoothing with Background

- 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 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

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

Scoring function

Where is *tf.idf* Weight?

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

$$= \sum_{i:f_{q_i,D}>0}\log\frac{((1-\lambda)\frac{f_{q_i,D}}{|D|} + \lambda\frac{c_{q_i}}{|C|})}{\lambda\frac{c_{q_i}}{|C|}} + \sum_{i=1}^{n}\log(\lambda\frac{c_{q_i}}{|C|})$$

$$\stackrel{rank}{=} \sum_{i:f_{q_i,D}>0}\log\left(\frac{((1-\lambda)\frac{f_{q_i,D}}{|D|}}{\lambda\frac{c_{q_i}}{|C|}} + 1\right)$$

- 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 estimate of

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

- and document score

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

Scoring function

Witten-Bell Smoothing

- α_D depends on unique "events" V in document

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

- Gives probability estimate of

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

- and document score

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

Scoring function

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
15	1	-13.75
15	0	-19.05
1	25	-12.99
0	25	-14.40

Summary of QL Model

- Query likelihood ranks documents by the probability that the author of D would have written the query Q.
  - P(Q | D)
- P(w | D) is a unigram language model.
  - Words are assumed independent.
- P(w | D) estimated by smoothed maximum-likelihood estimator.
  - Smoothing done by discounting, interpolating with background model, or backing off (not discussed).

Document-Likelihood Model

- Rank documents by the probability that the document could be generated by the query model (i.e. same topic)
- Given query, start with P(Q|D)
- Using Bayes' Rule

$$p(Q \mid D) = \frac{P(D \mid Q)P(Q)}{P(D)}$$

- Unigram model

$$\frac{P(D \mid Q)P(Q)}{P(D)} = \frac{P(Q)\prod_{i \in D} P(t_i \mid Q)}{\prod_{i \in D} P(t_i \mid C)}$$

Divergence Model

- Combine the advantages of query-likelihood and document-likelihood.
  - Estimate unigram query model P(w | Q) and unigram document model P(w | D).

$$H(Q \mid D) = -\sum_{w \in Q} P(w \mid Q) \log P(w \mid D)$$

Scoring function

  - H(Q | D) = number of bits needed to encode Q using D.
  - If P(w | Q) = $tf_{w,Q}$, equivalent to query-likelihood.

Discussion

- We only looked at unigram models for IR.
  - Unigram is a bad model of language.
  - Why would it work OK for IR?
- What is the best smoothing?
  - Don't make too many assumptions.
  - Don't smooth too much.
  - Dirichlet smoothing seems to provide the best "happy medium".
- Where is the relevance?
  - BIM and BM25 modeled P(R | D, Q).
  - There is no R in language models. What does that mean?