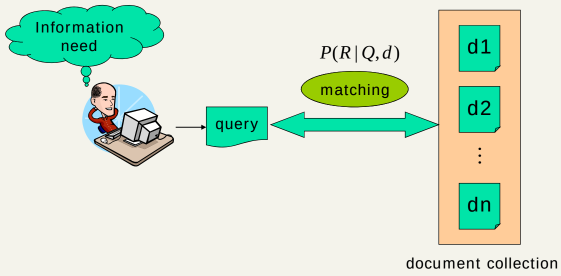
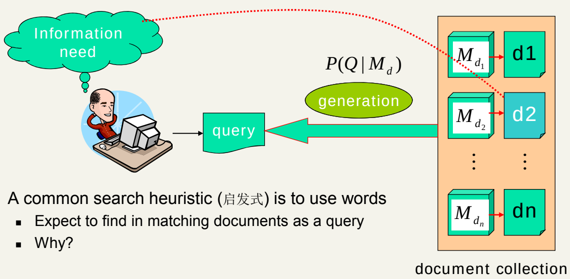
Information Retrieval

Language Models

Standard Probabilistic IR



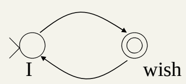
IR based on Language Model (LM)



- A common search heuristic (启发式) is to use words
  - Expect to find in matching documents as a query
  - Why?
- **The LM approach directly exploits that idea!**

Formal Language (Model)

- Traditional generative model: generates strings
  - Finite state machines (有限状态自动机) or regular grammars (正则语法)
  - etc.
- Example:



I wish
I wish I wish
I wish I wish I wish
I wish I wish I wish I wish
…………

*wish I wish

Stochastic (随机) Language Models

- Models *probability* of generating strings in the language (commonly all strings over alphabet ∑)

Model M

0.2	the
0.1	a
0.01	man
0.01	woman
0.03	said
0.02	likes
…	

the	man	likes	the	woman
0.2	0.01	0.02	0.2	0.01

multiply

P(s | M) = 0.00000008

Stochastic Language Models (续)

- Model *probability* of generating any string

Model M1

0.2	the
0.01	class
0.0001	sayst
0.0001	pleaseth
0.0001	yon
0.0005	maiden
0.01	woman

Model M2

0.2	the
0.0001	class
0.03	sayst
0.02	pleaseth
0.1	yon
0.01	maiden
0.0001	woman

the	class	pleaseth	yon	maiden
0.2	0.01	0.0001	0.0001	0.0005
0.2	0.0001	0.02	0.1	0.01

P(s|M2) > P(s|M1)

Stochastic Language Models (续)

- A statistical model for generating text
  - Probability distribution over strings in a given language

M

P (| M) = P (| M)

P (| M,)

P (| M,)

P (| M,)

Unigram and higher-order models

P ( )

= P ( ) P (|) P (|) P (|)

- Unigram Language Models
  P ( ) P ( ) P ( ) P ( )

Easy.
Effective!

- Bigram (generally, *n*-gram) Language Models
  P ( ) P (|) P (|) P (|)
- Other Language Models
  - Grammar-based models (PCFGs), etc.
    - Probably not the first thing to try in IR

Using Language Models in IR

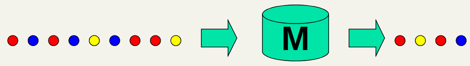
- Treat each document as the basis for a model
  - e.g., unigram sufficient statistics
- Rank document d based on **P(d | q)**

  P(d | q) = P(q | d) x P(d) / P(q)
  - P(q) is the same for all documents, so ignore
  - P(d) [the prior] is often treated as the same for all d
    - But we could use criteria like authority, length, genre
  - P(q | d) is the probability of q given d's model
- Very general formal approach

The fundamental problem of LMs

- Usually we don't know the model **M**
  - But have a sample of text representative of that model

- Estimate a language model from a sample
- Then compute the observation probability



Language Models for IR

- Language Modeling Approaches
  - Attempt to **model query generation process**
  - Documents are ranked by **the probability**
    - **that a query would be observed as a random sample from the respective document model**
    - Multinomial(多项式) approach

$$P(Q|M_D) = \prod_w P(w|M_D)^{q_w}$$

Retrieval based on probabilistic LM

- Treat the generation of queries as a random process.
- Approach

  1 Infer a language model for each document.

  2 Estimate the probability of generating the query according to each of these models.

  3 Rank the documents according to these probabilities.
  - Usually a unigram estimate of words is used

Query generation probability

- Ranking formula

$$p(Q,d) = p(d)p(Q \mid d)$$
$$\approx p(d)p(Q \mid M_d)$$

- The probability of producing the query given the language model of document d using MLE is:

$$\hat{p}(Q \mid M_d) = \prod_{t \in Q} \hat{p}_{ml}(t \mid M_d)$$
$$= \prod_{t \in Q} \frac{tf_{(t,d)}}{dl_d}$$

Unigram assumption:
Given a particular language model, the query terms occur independently

M_d : language model of document d

$tf_{(t,d)}$: raw tf of term t in document d

dl_d : total number of tokens in document d

Classic Problem

- Zero probability $p(t \mid M_d) = 0$
  - May not wish to assign a probability of zero to a document that is missing one or more of the query terms [gives conjunction semantics]
- General approach
  - A non-occurring term is possible, but no more likely than would be expected by chance in the collection.
  - If

$$tf_{(t,d)} = 0 \qquad p(t \mid M_d) = \frac{cf_t}{cs}$$

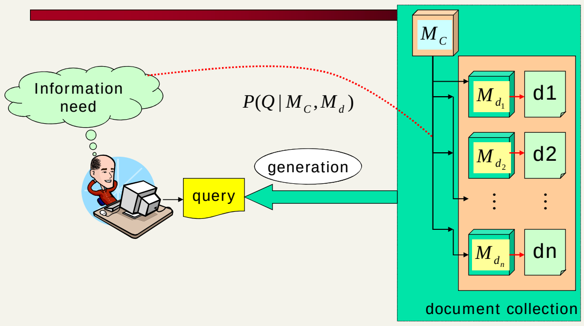
cf_t : raw count of term t in the collection

cs : raw collection size(total number of tokens in the collection)

Zero probabilities

- Need to smooth probabilities
  - Discount nonzero probabilities
  - Give some probability mass to unseen things
- There's a wide space of approaches to smoothing probability distributions to deal with this problem
  - such as
    - **adding 1, ½ or ε to counts**
    - **Dirichlet priors (Dirichlet 先验)**
    - **discounting**
    - **Interpolation (插值)**
      - [See FSNLP ch. 6 or CS224N if you want more]
- A simple idea that works well in practice
  - **use a mixture between the document multinomial and the collection multinomial distribution**

2-level model



Mixture model

- $P(w|d) = \lambda P_{mle}(w|M_d) + (1 - \lambda) P_{mle}(w|M_c)$
- Mixes the probability
  - from the document with the general collection frequency of the word.
- Correctly setting λ is very important
  - A high value of lambda suitable for short queries
    - makes the search "conjunctive-like"
  - A low value is more suitable for long queries
- Can tune λ to optimize performance
  - Perhaps make it dependent on document size
    - cf. Dirichlet prior or Witten-Bell smoothing

Basic mixture model summary

- General formulation of the LM for IR

$$p(Q,d) = p(d)\prod_{t \in Q}((1-\lambda)p(t) + \lambda p(t \mid M_d))$$

general language model

individual-document model

- The user has a document in mind and generates the query from this document.
- The equation represents the probability
  - that the document that the user had in mind was in fact this one.

Example

- Document collection (2 documents)

  d_1: Xerox reports a profit but *revenue* is *down*

  d_2: Lucent narrows quarter loss but *revenue* decreases further

- Model: MLE unigram from documents; $\lambda = ½$
- Query: *revenue down*
  - $P(Q|d_1) = [(1/8 + 2/16)/2] \times [(1/8 + 1/16)/2]$

    $= 1/8 \times 3/32 = 3/256$
  - $P(Q|d_2) = [(1/8 + 2/16)/2] \times [(0 + 1/16)/2]$

    $= 1/8 \times 1/32 = 1/256$
- Ranking: **$d_1 > d_2$**

LM vs. Prob. Model for IR

- The main difference is
  - whether "Relevance" figures explicitly in the model or not
- LM approach
  - Attempts to do away with modeling relevance
  - Assumes that documents and expressions of information problems are of the same type
  - Computationally
  - Intuitively

LM vs. Prob. Model for IR (续)

- Problems of basic LM approach
  - Assumption of equivalence between *document* and *information problem representation* is unrealistic
  - Very simple models of language
  - Relevance feedback is difficult to integrate
    - as are user preferences, and other general issues of relevance
  - Can't easily accommodate phrases, passages, Boolean operators

- Current extensions focus on
  - putting relevance back into the model
  - etc.