基于语言建模的检索模型

传统概率模型 vs 概率语言模型

- 传统概率模型
  - 需要对文档 d 与查询 q 的相关概率 $P(R = 1|q, d)$进行显式建模
- 概率语言模型
  - 首先对每篇文档 d 建模得到文档的概率语言模型 M_d
  - 然后按照模型生成查询q的概率$P(q|M_d)$的高低来对文档进行排序

本讲内容

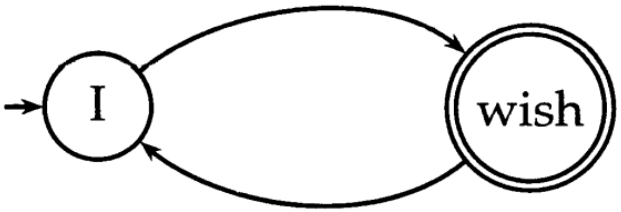
- 语言模型
  - 什么是概率语言模型？如何比较两个模型？
  - 怎样由文档生成语言模型？
  - 语言模型的种类
- 语言模型如何应用到IR中？
  - 查询似然模型(Query Likelihood Model)
  - 平滑的方法：线性插值LM
  - 扩展的LM方法

本讲内容

- 语言模型
  - 什么是概率语言模型？如何比较两个模型？
  - 语言模型的种类
  - 词的多项式分布
- 语言模型如何应用到IR中？
  - 查询似然模型(Query Likelihood Model)
  - 平滑的方法：线性插值LM
  - 扩展的LM方法

最简单的语言生成模型Generative Model

- 一个简单的有穷自动机及其生成语言中的一些字符串
  - → 指向的是自动机的初始状态
  - 双圈节点对应的是(可能的)终止状态



I wish

I wish I wish

I wish I wish I wish

I wish I wish I wish I wish I wish I wish

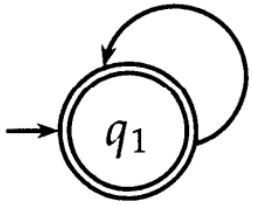
. . .

CANNOT GENERATE: wish I wish

- 如果每个节点都有一个生成不同词项的概率分布，便得到一个语言模型(Language Model)，或概率语言模型(Probabilistic LM)，或统计语言模型(Statistical LM)
  - 语言模型的概念本质上也是基于概率的。

有穷自动机→语言模型

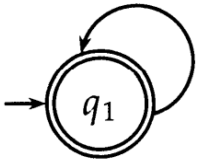
- 一个语言模型(LM)是从某词汇表上抽取的字符串到概率的一个映射函数。就是说，对于字母表Σ上的语言模型M有：$\sum_{s\in\Sigma^*} P(s)=1$
- 最简单的语言模型等价于一个仅仅包含一个节点的概率有穷自动机，只有一个生成不同词项的概率分布，因此有 $\sum_{t\in V} P(t)=1$
- 其中STOP不是词，而是表示自动机结束的一个标识符。



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

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

一个概率LM的例子



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

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

P(frog said that toad likes frog)=
(0.01×0.03×0.04×0.01×0.02×0.01)
× (0.8×0.8×0.8×0.8×0.8×0.8×0.2)
≈ 0.000 000 000 001 573

词项发射概率

生成每个词后继续前进或停止的转移概率

- 假定停止概率是固定的，因此，它不会影响文档的排序 。
- 因此可以不考虑停止概率，但形式上得到的结果将不再是概率，而只是概率的部分项。
- 课程中后续内容将忽略停止概率

语言模型的比较

比较两个模型，可计算似然比(likelihood ratio)，即将其中一个模型的数据生成概率除以另外一个模型的数据生成概率。

模式 M_1		模式 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
…	…	…	…

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

的部分概率赋值

$P(s|M_1) = 0.000\ 000\ 000\ 000\ 48$ ，

$P(s|M_2) = 0.000\ 000\ 000\ 000\ 000\ 384$。

$P(s|M_1) > P(s|M_2)$说明s由M_1生成的可能性大

这里是对概率求积，但是通常在概率应用中，往往采用对数求和的计算方法

本讲内容

- 语言模型
  - 什么是概率语言模型？如何比较两个模型？
  - 语言模型的种类
  - 词的多项式分布
- 语言模型如何应用到IR中？
  - 查询似然模型(Query Likelihood Model)
  - 平滑的方法：线性插值LM
  - 扩展的LM方法

对于词项序列如何求解其生成的概率值？

- 根据链式规则将一系列事件的概率分解成多个连续的事件概率之积，每个概率是每个事件基于其历史事件的条件概率。
- 计算公式如下：
  - $P(t_1t_2t_3t_4) = P(t_1)P(t_2|t_1)P(t_3|t_1t_2)P(t_4|t_1t_2t_3)$

语言模型的种类*n*-gram

- 一元语言模型(Unigram LM)：也称上下文无关语言模型，是最简单的语言模型，去掉所有条件概率中的条件来独立地估计每个词项的概率。
  - $P_{\text{uni}}(t_1t_2t_3t_4) = P(t_1)P(t_2)P(t_3)P(t_4)$
  - 词袋模型 Bag of words
- 二元语言模型(Bigram LM)，即计算条件概率时只考虑前一个词项的出现情况：
  - $P_{\text{bi}}(t_1t_2t_3t_4) = P(t_1)P(t_2|t_1)P(t_3|t_2)P(t_4|t_3)$
- 三元语言模型(Trigram LM)
- …
- Unigram→ Bigram→…→*n*-gram

  还有其他更复杂的种类，本课程不介绍

本讲内容

- 语言模型
  - 什么是概率语言模型？如何比较两个模型？
  - 语言模型的种类
  - 词的多项式分布
- 语言模型如何应用到IR中？
  - 查询似然模型(Query Likelihood Model)
  - 平滑的方法：线性插值LM
  - 扩展的LM方法

由一元语言模型产生一个文档d的概率？

- 词的多项式分布
- 一元语言模型会给有序的词项序列赋予概率。当然，这些词项按照其他次序出现的概率也等于这个概率。因此，实际上这相当于词项存在一个多项式分布。也可以称为多项式模型。

$$P(d)=\frac{L_d!}{tf_{t_1,d}!\,f_{t_2,d}!\cdots f_{t_M,d}!}P(t_1)^{tf_{t_1,d}}P(t_2)^{tf_{t_2,d}}\cdots P(t_M)^{tf_{t_M,d}}$$

  - $L_d=\sum_{1\le i\le M}tf_{t_i,d}$ 是文档的长度(即词条的总个数)
  - M是词典的大小

多项式分布的概率公式

$$P(X_1=x_1,...,X_k=x_k)=\begin{cases}\dfrac{n!}{x_1!...x_k!}P(1)^{x_1}\cdots P(k)^{x_k} & \text{when } \sum_{i=1}^{k}x_i=n\\ 0 & \text{otherwise}\end{cases}$$

k个状态，第1个状态发生概率$P(1)$，发生了x_1次。总共发生了n次

本讲内容

- 语言模型
  - 什么是概率语言模型？如何比较两个模型？
  - 语言模型的种类
  - 词的多项式分布
- 语言模型如何应用到IR中？
  - 查询似然模型(Query Likelihood Model)
  - 平滑的方法：线性插值LM
  - 扩展的LM方法

语言模型如何应用到IR中？

- IR中使用LM的问题
  - N个文档，各自有一个语言模型，给定一个查询，求查询与哪个文档相关度最高？
- 对比问题
  - 问题：设有N个作者，每人有一篇文章，对于不在上述N篇文章中的一篇新文档q，问最有可能是哪个作者写的？
  - 一个解决思路：根据每个作者写的文章，总结出作者的写作风格，然后根据写作风格来判断q与谁的风格最近。
  - 一种可能的思路：把相关度看成是每篇文档对应的语言模型下生成该查询的可能性

总体分布&抽样

- 文档的模型(风格)实际上是某种总体分布
- 文档和查询都是该总体分布下的一个抽样样本实例
- 根据文档，估计文档的模型，即求出该总体分布(一般假设某种总体分布，然后求出其参数)
- 然后计算该总体分布下抽样出查询的概率

文档→总体分布→查询

查询似然模型(query likelihood model)

- 每篇文档d构建其对应的语言模型M_d
- 将文档按照其与查询相关的似然$P(d|q)$排序
  - $P(d|q) = P(q|d)\ P(d)/P(q)$

文档无关

均匀分布

- 最后会按照$P(q|d)$进行排序，它是在文档 d 对应的语言模型M_d下生成 q 的概率
- 问题：已知样本d，求其模型M_d的参数$P(w|M_d)$

- 因此，IR 中的语言建模方法实际上是在对查询的生成过程进行建模：
  - 首先每篇文档d对应一个文档模型M_d
  - 然后计算查询被视为每个文档模型的随机抽样样本的概率
  - 最后根据这些概率对文档排序。

$$P(q \mid M_d) = K_q \prod_{t \in V} P(t \mid M_d)^{tf_{t,d}}$$

- $K_q = \dfrac{L_d!}{tf_{t_1,d}!\, f_{t_2,d}!\cdots f_{t_M,d}!}$ 是查询 q 的多项式系数，对于某个特定的查询，是一个常数，可以忽略

- 基于语言模型(LM)的检索，将查询的生成看成一个随机过程。**具体方法**：
  - (1) 对每篇文档d_i推导出其LM M_{di}；
  - (2) 估计查询在每个文档 d_i的LM下的生成概率 $P(q|M_{di})$；
  - (3) 按照上述概率对文档进行排序。
- 上述模型的直观意义是，用户脑子里有一篇原型文档，然后按照该文档中的词语用法来生成查询。通常，用户对感兴趣的文档中可能出现的词项有一些合理的想法，然后他们会选择那些能够区分其他文档的查询项构成查询。

查询生成概率的估计

- 每篇文档d构建其对应的语言模型M_d
- 采用最大似然估计(Maximum Likelihood Estimation，MLE)：使得观察样本出现概率(似然)最大的估计

$$\hat{P}(q \mid M_d) = \prod_{t \in q} \hat{P}_{\text{mle}}(t \mid M_d) = \prod_{t \in q} \frac{tf_{t,d}}{L_d}$$

  - L_d是d中的词条数目

小结：查询似然模型

- 查询似然模型(query likelihood model, QLM)
  - 目标：将文档按照其与查询相关的似然$P(d|q)$ 排序
  - 实现目标的途径：按照$P(q|d)$进行排序
- 具体的方法是：
  - (1) 对每篇文档推导出其LM
  - (2) 估计查询在每个文档d_i的LM下的生成概率$P(q|M_{di})$

$$P(q \mid M_d) = K_q \prod_{t \in V} P(t \mid M_d)^{tf_{t,d}}$$

估计

$$\hat{P}(q \mid M_d) = \prod_{t \in q} \hat{P}_{\text{mle}}(t \mid M_d) = \prod_{t \in q} \frac{tf_{t,d}}{L_d}$$

  - (3) 按照上述概率对文档进行排序

本讲内容

- 语言模型
  - 什么是概率语言模型？如何比较两个模型？
  - 语言模型的种类
  - 词的多项式分布
- 语言模型如何应用到IR中？
  - 查询似然模型(Query Likelihood Model)
  - 平滑的方法：线性插值LM
  - 扩展的LM方法

平滑的方法：线性插值LM

$$\hat{P}(q \mid M_d) = \prod_{t \in q} \hat{P}_{\text{mle}}(t \mid M_d) = \prod_{t \in q} \frac{tf_{t,d}}{L_d}$$

- 如果$\hat{P}_{\text{mle}}(t \mid M_d) = 0$，怎么办？
- 因此，需要对文档LM的概率进行**平滑(Smoothing)**，即对出现事件的概率结果进行折扣，并对未出现的词的概率赋予一定的值。
- 对概率分布进行平滑方法的有很多
  - 在事件的观察数目上加某个数字(1、1/2或一个很小的α)，然后对概率分布重新进行归一化的平滑方法。
  - 线性插值LM(linear interpolation LM)

线性插值(Linear Interpolation) LM

- 在一般的参照概率分布中，文档中未出现的查询项都可能在查询中出现，它的概率在某种程度上接近但不可能超过在整个文档集中偶然出现的概率。也就是说，如果$tf_{t,d}=0$，那么有

$$\hat{P}_{\text{mle}}(t \mid M_d) \le \frac{cf_t}{T}$$

  - cf_t是t在整个文档集的出现次数，T是所有文档集中词条的个数

- 实际效果较好的简单方法：将基于文档的多项式分布和基于全部文档集估计出的多项式分布相混合，即

$$\hat{P}(t \mid d) = \lambda \hat{P}_{\text{mle}}(t \mid M_d) + (1-\lambda)\hat{P}_{\text{mle}}(t \mid M_c)$$

  - $0<\lambda<1$，M_c是基于全部文档集构造的LM

线性插值LM示例

- 在语言建模的IR模型下，查询q的检索排序函数定义如下

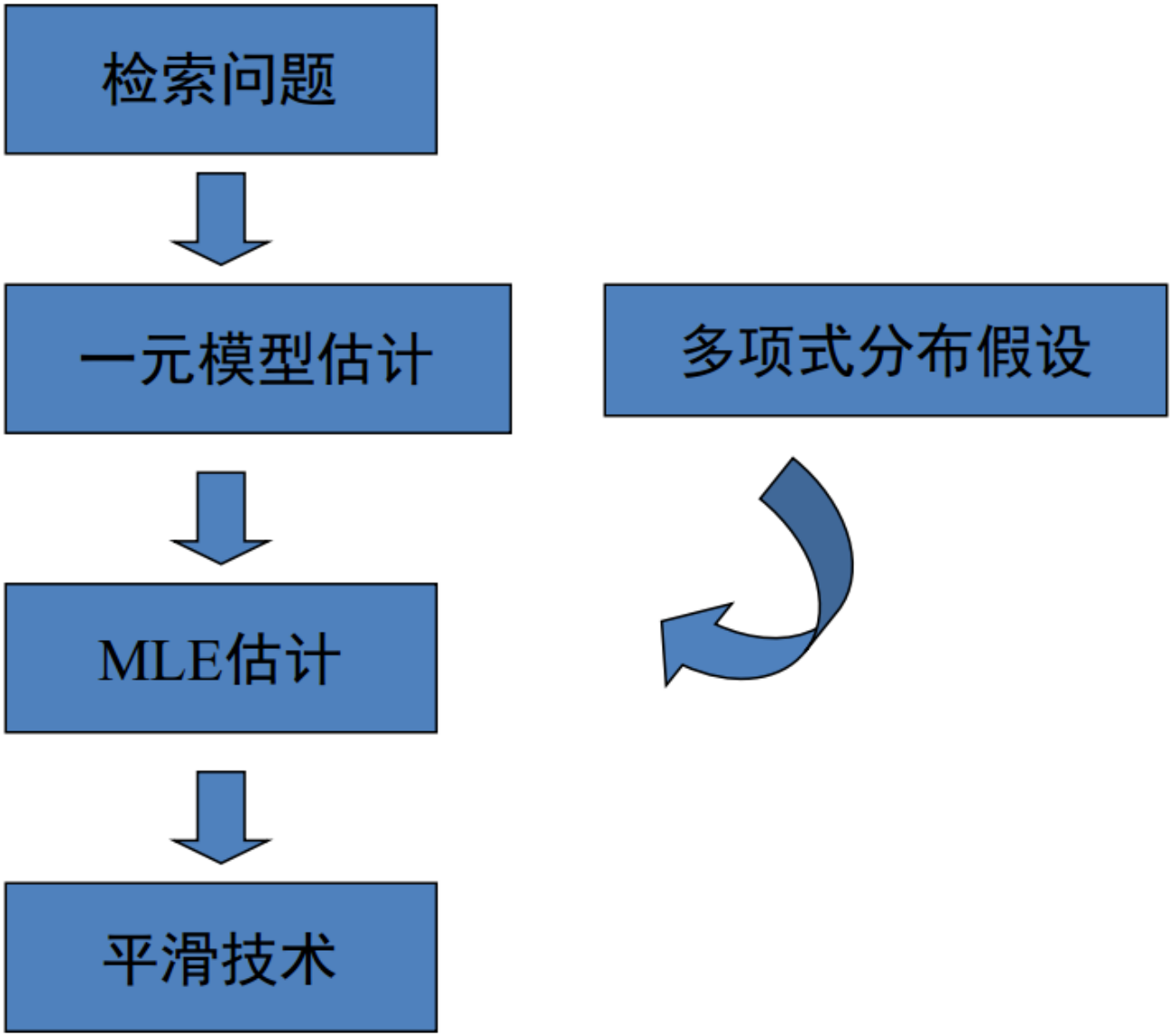
$$P(d \mid q) \propto P(d) \prod_{t \in q} [\lambda P(t \mid M_d) + (1-\lambda) P(t \mid M_c)]$$

- 例12-3 假定某文档集中包含如下两篇文档：
  - d_1: Xyzzy reports a profit but revenue is down (8)
  - d_2: Quorus narrows quarter loss but revenue decreases further (8)
  - 混合参数$\lambda = 1/2$。假定查询为revenue down，则有

$$P(q \mid d_1) = [\frac{\frac{1}{8} + \frac{2}{16}}{2}] \times [\frac{\frac{1}{8} + \frac{1}{16}}{2}] = \frac{1}{8} \times \frac{3}{32} = \frac{3}{256}$$

$$P(q \mid d_2) = [\frac{\frac{1}{8} + \frac{2}{16}}{2}] \times [\frac{\frac{0}{8} + \frac{1}{16}}{2}] = \frac{1}{8} \times \frac{1}{32} = \frac{1}{256}$$

小结：QLM(Query Likelihood Model)模型

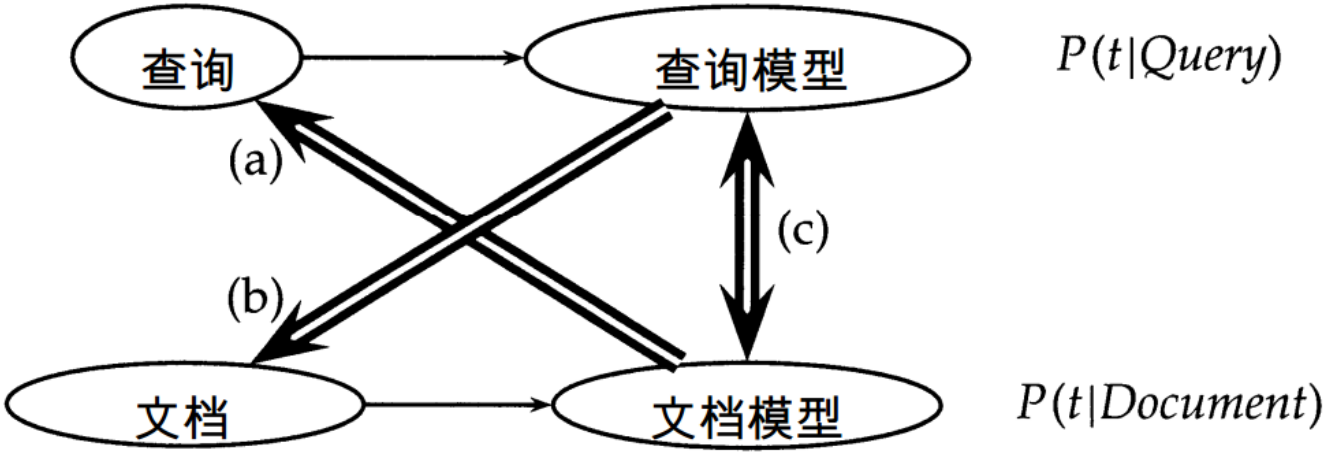


本讲内容

- 语言模型
  - 什么是概率语言模型？如何比较两个模型？
  - 语言模型的种类
  - 词的多项式分布
- 语言模型如何应用到IR中？
  - 查询似然模型(Query Likelihood Model)
  - 平滑的方法：线性插值LM
  - 扩展的LM方法

扩展的LM 方法

- a 查询似然类：文档建模，计算查询的似然
  - 例子--基本**QLM**模型、 翻译模型等
- b 文档似然类：查询建模，计算文档的似然
  - 例子--**BIM**模型、相关性模型**(Relevance**模型**)**等
- c 模型比较类：文档建模、查询建模，计算两个模型的距离，**KL**距离模型



IR 中使用统计语言建模的 3 种方式：(a) 查询似然 (b) 文档似然 (c) 模型比较

KL距离模型

- 通过计算查询模型M_q和文档模型M_d的KL距离(Kullback-Leibler divergence)来对返回文档d的风险进行建模 $$R(d;q) = KL(M_d \| M_q) = \sum_{t \in V} P(t \mid M_q) \log \frac{P(t \mid M_q)}{P(t \mid M_d)}$$
- KL距离源自信息论的一个非对称距离度量方法，主要度量的是概率分布M_d对M_q建模的无效程度。
- Lafferty和Zhai(2001)给出的结果表明基于模型对比的方法比查询似然和文档似然的方法都好。
- KL作为排序函数的缺点：最后的得分在查询之间没有可比性。
  - 对于ad hoc检索来说没有关系，但是对于一些其他应用(如话题跟踪)影响很大。

翻译模型

- 基本的LM方法没有考虑表达方式不同的问题，比如一义多词或者查询语言和文档语言间的偏差问题。
- 翻译模型可以通过翻译的方法，让不在文档中的词项转变为其近义词项后出现在查询中。
- 这种方法也为跨语言IR提供了基础。
- 假定翻译模型可以通过一个词项间的条件概率分布 $T(.|.)$来表示，则基于翻译的查询生成模型定义为：

$$P(q \mid M_d) = \prod_{t \in q} \sum_{v \in V} P(v \mid M_d) T(t \mid v)$$

  - 其中，$P(v|M_d)$是基本的文档语言模型
  - $T(t|v)$表示翻译概率。翻译模型往往基于单独的资源来构建(比如传统的同义词词典或者双语词典)。当然，如果存在一些文本片段很自然地解释或者概括了其他文本，那么也可以基于文档集来构建翻译模型。