

# 现代信息检索

## Modern Information Retrieval

### 第9讲 基于统计语言建模的IR模型

### Statistical Language Models for IR

# 提纲

- ① 上一讲回顾
- ② 统计语言模型(SLM)
- ③ 基于SLM的IR模型
- ④ SLMIR模型讨论

# 提纲

- ① 上一讲回顾
- ② 统计语言模型(SLM)
- ③ 基于SLM的IR模型
- ④ SLMIR模型讨论

# 概率检索模型

- 概率检索模型是通过概率的方法将查询和文档联系起来
  - 定义3个随机变量 $R$ 、 $Q$ 、 $D$ ：相关度 $R=\{0,1\}$ ，查询 $Q=\{q_1, q_2, \dots\}$ ，文档 $D=\{d_1, d_2, \dots\}$ ，则可以通过计算条件概率 $P(R=1|Q=q, D=d)$ 来度量文档和查询的相关度。
- 概率模型包括一系列模型，如Logistic Regression(回归)模型及最经典的二值独立概率模型BIM、BM25模型等等(还有贝叶斯网络模型)。
- 1998出现的基于统计语言建模的信息检索模型本质上也是概率模型的一种。

# 概率排序原理(PRP)

---

- 简单地说：如果文档按照与查询的相关概率大小返回，那么该返回结果是所有可能获得结果中效果最好的。
- 严格地说：如果文档按照与查询的相关概率大小返回，而这些相关概率又能够基于已知数据进行尽可能精确的估计，那么该返回结果是所有基于已知数据获得的可能的结果中效果最好的。

# 几种概率检索模型

---

- 基于Logistic回归的检索模型
- 经典的二值独立概率模型BIM
- 经典的BM25模型 (BestMatch25)

# Logistic 回归IR模型

- 基本思想：为了求 $Q$ 和 $D$ 相关的概率 $P(R=1|Q,D)$ ，通过定义多个特征函数 $f_i(Q,D)$ ，认为 $P(R=1|Q,D)$ 是这些函数的组合。
- Cooper等人提出一种做法\*：定义 $\log(P/(1-P))$ 为多个特征函数的线性组合。则 $P$ 是一个Logistic函数，即：

$$\log \frac{P}{1-P} = \beta_0 + \sum_i \beta_i f_i(Q,D)$$

$$P = \frac{1}{1 + e^{-\beta_0 - \sum_i \beta_i f_i(Q,D)}}$$

\*William S. Cooper , Fredric C. Gey , Daniel P. Dabney, Probabilistic retrieval based on staged logistic regression, Proceedings of ACM SIGIR'92, p.198-210, June 21-24, 1992, Copenhagen, Denmark

# BIM模型(续)

- 对每个 $Q$ 定义排序(Ranking)函数RSV( $Q, D$ ):

$$\log \frac{P(R=1|D)}{P(R=0|D)} = \log \frac{P(D|R=1)P(R=1)/P(D)}{P(D|R=0)P(R=0)/P(D)}$$
$$\propto \log \frac{P(D|R=1)}{P(D|R=0)}$$

其中,  $P(D|R=1)$ 、 $P(D|R=0)$ 分别表示在相关和不相关情况下生成 $D$ 的概率。Ranking函数显然是随着 $P(R=1|D)$ 的增长而增长。



# 两种常用的文档生成的总体分布

- 多元贝努利分布(Multi-variate Bernoulli distribution)
  - 词项词典大小为 $M$ ， $M$ 个不规则硬币分别对应 $M$ 个词项，第 $i$ 个硬币朝上的概率为 $p_i$
  - 假设 $M=4$ (四个词项分别为 I you can fly),  $p_1=0.7$ ,  $p_2=0.4$ ,  $p_3=0.1$ ,  $p_4=0.05$
  - 则:  $P(\text{I can fly fly})=0.7*(1-0.4)*0.1*0.05$
  - 多元贝努利分布不考虑出现位置
  - 多元贝努利分布考虑出现和不出现



# 两种常用的文档生成的总体分布(续)

- 多项式分布(Multinomial distribution)
  - 词项大小为M, 某个不规则骰子共有M个面, 每个面对应一个词项(假设每次抛掷必有某个面稳定朝上或下), 第*i*个面朝上的概率为 $p_i$
  - 假定M=4 (四个词项分别为 I you can fly),  $p_1=0.4$ ,  $p_2=0.3$ ,  $p_3=0.2$ ,  $p_4=0.1$
  - 则:  $P(\text{I can fly fly})=P(X_1=1, X_2=0, X_3=1, X_4=2)$   
 $=C \cdot 0.4 \cdot 0.2 \cdot 0.1 \cdot 0.1$
  - 其中 $C = C_4^1 \times C_3^1 \times C_2^2 = 12$
  - 多项式分布考虑词项的多次出现
  - 多项式分布不考虑词项的不出现
  - 多项式分布同样不考虑词项的出现位置和次序



# BIM模型(续)

将 $D$ 看成  $\bigwedge_{t_i \in D} t_i \bigwedge_{t_j \notin D} \bar{t}_j$

$$\begin{aligned} P(D | R = 1) &= \prod_{t_i \in D} P(t_i | R = 1) \prod_{t_i \notin D} P(\bar{t}_i | R = 1) \\ &= \prod_{t_i} p_i^{e_i} (1 - p_i)^{1-e_i}, \text{if } t_i \in D \text{ then } e_i = 1, \text{else } e_i = 0 \end{aligned}$$

$$p_i = P(t_i | R = 1)$$

$$q_i = P(t_i | R = 0)$$

$$\begin{aligned} P(D | R = 0) &= \prod_{t_i \in D} P(t_i | R = 0) \prod_{t_i \notin D} P(\bar{t}_i | R = 0) \\ &= \prod_{t_i} q_i^{e_i} (1 - q_i)^{1-e_i}, \text{if } t_i \in D \text{ then } e_i = 1, \text{else } e_i = 0 \end{aligned}$$

**注：** $P(t_i | R=1)$ 表示在相关情况下， $t_i$ 出现在文档中的概率(也就是说某个、或者某几个 $P(t_i | R=1)$ 可以为1)，注意：不是在相关文档集合中出现的概率，因此所有 $P(t_i | R=1)$ 的总和不等于1。这个可以和前面抛硬币的过程对照一下就明白了。

# $p_i$ $q_i$ 参数的计算

理想情况下，可以将整个文档集合根据是否和查询相关、是否包含 $t_i$ 分成如下四个子集合，每个集合的大小已知。

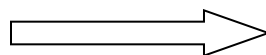
	相关 $R_i$ (100)	不相关 $N-R_i$ (400)
包含 $t_i$ $n_i$ (200)	$r_i$ (35)	$n_i - r_i$ (165)
不包含 $t_i$ $N-n_i$ (300)	$R_i - r_i$ (65)	$N - R_i - n_i + r_i$ (235)

其中， $N$ 、 $n_i$ 分别是总文档以及包含 $t_i$ 的文档数目。 $R_i$ 、 $r_i$ 分别是相关文档及相关文档中包含 $t_i$ 的文档数目。括号中列举的数值是给出的一个总文档数目为500的计算例子。则：

$$p_i = \frac{r_i}{R_i} = \frac{35}{100} = 0.35$$

$$q_i = \frac{n_i - r_i}{N - R_i} = \frac{165}{400} = 0.413$$

引入平滑因子



$$p_i = \frac{r_i + 0.5}{R_i + 1}$$

$$q_i = \frac{n_i - r_i + 0.5}{N - R_i + 1}$$

## $p_i$ $q_i$ 参数的计算(续)

- 由于真实情况下，对于每个查询，无法事先得到相关文档集和不相关文档集，所以无法使用理想情况下的公式计算，因此必须进行估计
- 有多种估计方法
  - 初始检索：第一次检索之前的估计
  - 基于检索结果：根据上次检索的结果进行估计

# $p_i$ $q_i$ 参数的计算(续)

- 初始情况：检索初始并没有相关和不相关文档集合，此时可以进行假设： $p_i$ 是常数， $q_i$ 近似等于term  $i$ 在所有文档集合中的分布 (假定相关文档很少， $R_i=r_i=0$ )

$$p_i = 0.5$$

$$q_i = \frac{n_i}{N}$$

$$\sum_{t_i \in q \cap d} \log \frac{p_i / (1 - p_i)}{q_i / (1 - q_i)} = \sum \log \frac{N - n_i}{n_i}$$



因此，BIM在初始假设情况下，其检索公式实际上相当于对所有同时出现在 $q$ 和 $d$ 中的term的IDF的求和

# Okapi BM25: 一个非二值模型

$$RSV(Q, D) = \sum_{t_i \in D \cap Q} W_i^{IDF} \cdot \frac{(k_1 + 1)tf_{ti,D}}{k_1((1-b) + b \times (L_D / L_{ave})) + tf_{ti,D}} \cdot \frac{(k_3 + 1)tf_{ti,Q}}{k_3 + tf_{ti,Q}}$$

$$W_i^{IDF} = \log \frac{N - n_i + 0.5}{n_i + 0.5}$$

# 本讲内容

---

- (统计)语言模型
- 基于统计语言建模的IR模型
  - (基本)查询似然模型
  - 一些扩展的模型



# 提纲

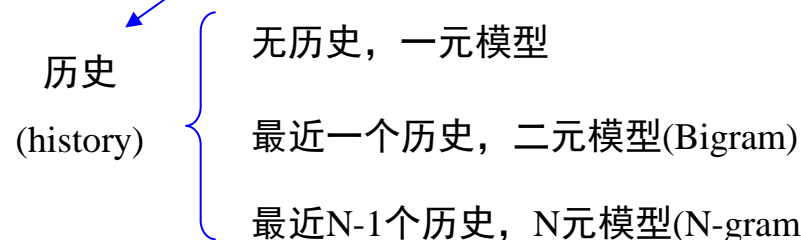
- ① 上一讲回顾
- ② 统计语言模型
- ③ 基于统计建模的IR模型
- ④ SLMIR模型讨论

# 统计语言模型

## (Statistical Language Modeling, SLM)

- SLM广泛使用于语音识别和统计机器翻译领域，利用概率统计理论研究语言。
  - 规则方法：词、句、篇章的生成比如满足某些规则，不满足该规则就不应存在。
  - 统计方法：任何语言片断都有存在的可能，只是可能性大小不同
- 对于一个文档片段 $d=w_1w_2...w_n$ ，统计语言模型是指概率 $P(w_1w_2...w_n)$ 求解，根据Bayes公式，有

$$P(w_1w_2...w_n) = P(w_1)P(w_2...w_n | w_1) = ... = P(w_1) \prod_{i=2}^n P(w_i | w_{i-1}...w_1)$$



# 类比：打扑克中的出牌策略

只根据当前牌出牌，一元模型；

根据上一轮牌出牌，二元模型；

.....



# 不同模型的例子

- 一元模型(unigram):  $P(w_1 w_2 w_3 w_4) = P(w_1)P(w_2)P(w_3)P(w_4)$
- 二元模型(bigram):  $P(w_1 w_2 w_3 w_4) = P(w_1)P(w_2 | w_1)P(w_3 | w_2)P(w_4 | w_3)$ 
  - 一阶马尔科夫链
- 三元模型(trigram):  $P(w_1 w_2 w_3 w_4) = P(w_1)P(w_2 | w_1)P(w_3 | w_1 w_2)P(w_4 | w_2 w_3)$
- 对于 $n$ -gram,  $n$ 越大, 则模型越复杂, 估计的参数(即估计的概率)也越多。当然, 当数据量足够大的情况下, 模型阶数越高对片段概率的计算也越准确。

# 课堂思考

- 设词典大小为 $M$ ，试估计 $N$ 元模型要估计的参数(概率)空间大小。

$$P(w_1 w_2 w_3 w_4) = P(w_1)P(w_2)P(w_3)P(w_4)$$

$$P(w_1 w_2 w_3 w_4) = P(w_1)P(w_2 | w_1)P(w_3 | w_2)P(w_4 | w_3)$$

$$P(w_1 w_2 w_3 w_4) = P(w_1)P(w_2 | w_1)P(w_3 | w_1 w_2)P(w_4 | w_2 w_3)$$

- 估计的参数数目为： $M + M^2 + \dots + M^N = (M^{N+1} - M) / (M - 1)$
- 假定 $M=1000$ ， $N=4$ ，则需要估计约 $10^{12}=1$ 万亿个参数，参数空间巨大！

# SLM的一个应用例子

- 拼音输入法(以下例子中将字看成语言单位):
  - 输入zhong guo ke xue yuan, 到底是: 种过科雪园? 重果可薛原? 还是 中国科学院? .....
  - 一种利用SLM的解决思路: 计算  $P(\text{种过科雪园})$   $P(\text{重果可薛原})$   $P(\text{中国科学院})$ , 看谁大! (为简单起见, 这里计算没有考虑拼音, 实际上是计算  $P(\text{种过科雪园} | \text{zhong guo ke xue yuan})$ )
- 一元模型(Unigram)\*:
  - $P(\text{种过科雪园}) = P(\text{种}) P(\text{过}) P(\text{科}) P(\text{雪}) P(\text{园})$
  - $P(\text{重果可薛原}) = P(\text{重}) P(\text{果}) P(\text{可}) P(\text{薛}) P(\text{原})$
  - $P(\text{中国科学院}) = P(\text{中}) P(\text{国}) P(\text{科}) P(\text{学}) P(\text{院})$
  - 训练: 在训练语料库中估计以上各  $P(X)$  的值
- 课堂思考: 一元模型存在的问题?

# SLM的一个应用例子(续)

- 二元模型(Bigram):  $P(\text{中国科学院}) = P(\text{中})P(\text{国}|\text{中})P(\text{科}|\text{国})P(\text{学}|\text{科})P(\text{院}|\text{学})$ , 等价于一阶马尔科夫链(Markov Chain)
- 三元模型(Trigram):  $P(\text{中国科学院}) = P(\text{中})P(\text{国}|\text{中})P(\text{科}|\text{中国})P(\text{学}|\text{国科})P(\text{院}|\text{科学})$
- 根据语料, 估计所使用模型的参数, 然后在搜索空间中搜索概率最大的语言片段。

# SLM的参数估计

---

- 理论上说，在数据充足的情况下，利用更多的历史(高阶)的模型更准确，但是总计算量也越大
- 数据规模总是有限的，即用于训练模型参数的语料存在稀疏性(Data Sparseness，即某参数在训练语料中没有出现)问题。如二元模型中，在训练语料中恰巧没有出现“国科”组合。



# 数据稀疏性

- 数据稀疏性导致零概率问题，上述稀疏性情况下，如果直接计算，那么 $P(\text{中国科学院})=0$ ，但是在训练集上不出现的事件并不代表在新的语料上不出现。
- SLM的一个重要工作就是进行平滑(Smoothing)：重新分配概率，即使没出现的事件也会赋予一个概率。

## 另一个角度看语言模型

我们可以把一个有穷状态自动机(finite state automaton)看成一个确定性语言模型(deterministic language)



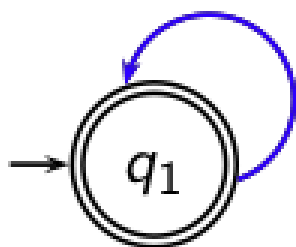
上述模型可以生成片段 “I wish I wish I wish I wish ...”  
但是不能生成片段 “wish I wish” 或 “I wish I”

如果上述自动机是带有概率的，则是概率语言模型(probabilistic LM，也称统计语言模型SLM)

# 提纲

- ① 上一讲回顾
- ② 统计语言模型 (SLM)
- ③ 基于SLM的IR模型
- ④ SLMIR模型讨论

# 一个概率语言模型的例子



$w$	$P(w q_1)$	$w$	$P(w q_1)$
STOP	0.2	toad	0.01
the	0.2	said	0.03
a	0.1	likes	0.02
frog	0.01	that	0.04
		...	...

单状态概率有穷状态自动机——一元语言模型——状态发射概率分布如右表。其中STOP不是词，而是表示自动机结束的一个标识符。这样，概率

$$P(\text{frog said that toad likes frog STOP}) = 0.01 \cdot 0.03 \cdot 0.04 \cdot 0.01 \cdot 0.02 \cdot 0.01 \cdot 0.02 = 0.000000000000048$$

# 两个不同的语言模型

language model of  $d_1$

$w$	$P(w .)$	$w$	$P(w .)$
STOP	.2	toad	.01
the	.2	said	.03
a	.1	likes	.02
frog	.01	that	.04
		...	...

language model of  $d_2$

$w$	$P(w .)$	$w$	$P(w .)$
STOP	.2	toad	.02
the	.15	said	.03
a	.08	likes	.02
frog	.01	that	.05
		...	...

string = frog said that toad likes frog STOP

则  $P(\text{string} | M_{d_1}) = 0.01 \cdot 0.03 \cdot 0.04 \cdot 0.01 \cdot 0.02 \cdot 0.01 \cdot 0.02 = 0.00000000000048 = 4.8 \cdot 10^{-12}$

$P(\text{string} | M_{d_2}) = 0.01 \cdot 0.03 \cdot 0.05 \cdot 0.02 \cdot 0.02 \cdot 0.01 \cdot 0.02 = 0.00000000000120 = 12 \cdot 10^{-12}$        $P(\text{string} | M_{d_1}) < P(\text{string} | M_{d_2})$

因此, 相对于  $d_1$ , 文档  $d_2$  与字符串 “frog said that toad likes frog STOP” 更相关

# 统计语言建模IR模型(SLMIR)

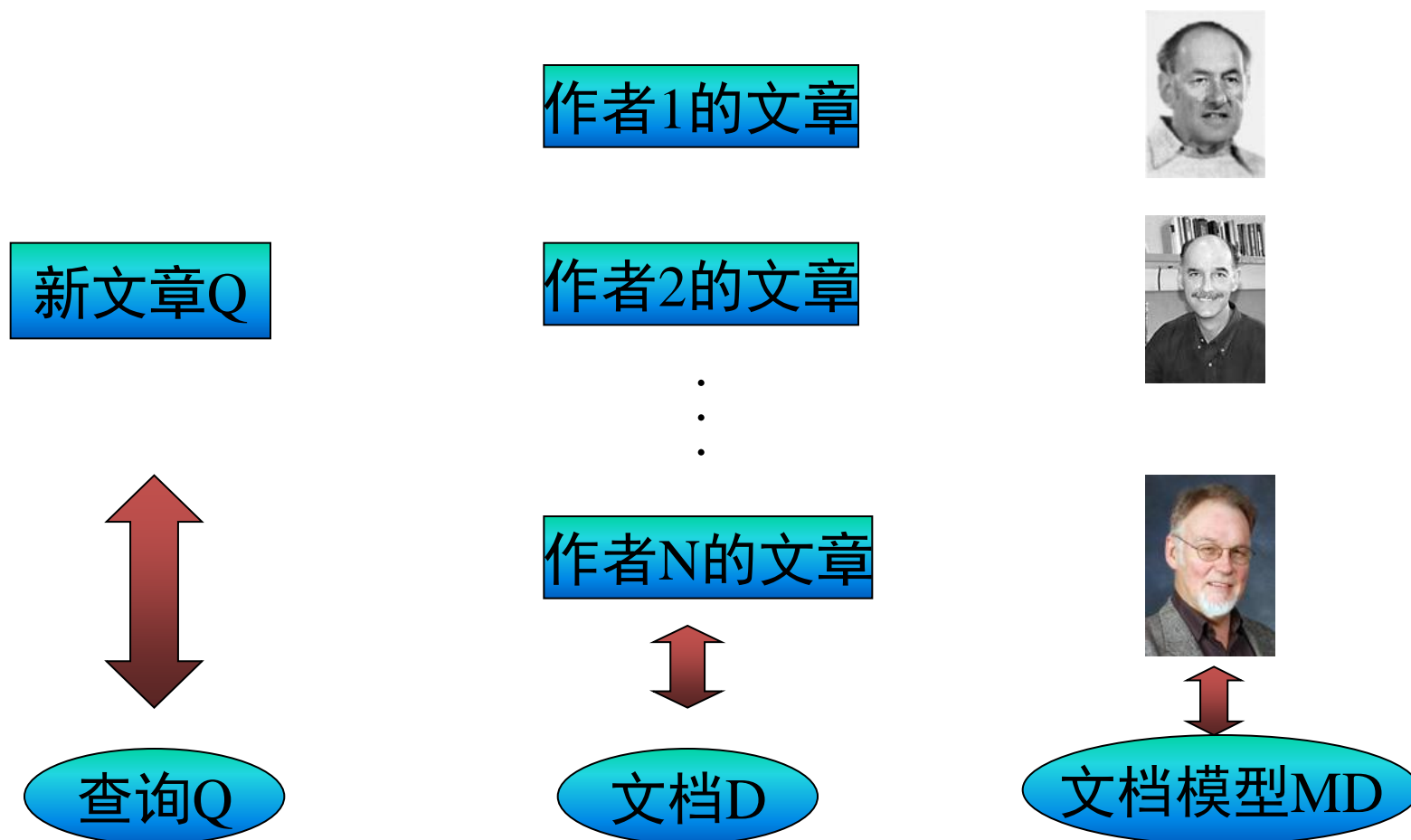
- 马萨诸塞大学(University of Massachusetts, UMass)大学Ponte、Croft等人于1998年提出。随后又发展了出了一系列基于SLM的模型。代表系统Lemur。
  - **查询似然模型**：把相关度看成是每篇文档对应的语言下生成该查询的可能性
  - **翻译模型**：假设查询经过某个噪声信道变形成某篇文章，则由文档还原成该查询的概率(翻译模型)可以视为相关度
  - **KL距离模型**：查询对应某种语言，每篇文档对应某种语言，查询语言和文档语言的KL距离作为相关度量
- 本讲义主要介绍查询似然模型

# 从一个问题开始

---

- 课堂思考题：设有 $N$ 个作者，每人都有一篇文章，对于不在上述 $N$ 篇文章中的一篇新文档 $Q$ ，请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，总结出每个作者的写作风格，然后再根据写作风格来判断 $Q$ 与谁的风格最近。

# 和检索的类比





# 总体分布&抽样

---

- 文档的模型(风格)实际上是某种**总体分布**
- （待评分）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

# 查询似然模型(Query Likelihood Model)

- 模型推导：

$$P(D|Q) = \frac{P(Q|D)P(D)}{P(Q)} \propto P(Q|D)P(D)$$

- 文档 $D$ 的先验分布 $P(D)$ 假定为均匀分布，则这一项可以去掉。
- $P(D)$ 也可以采用某个与查询无关的量，如PageRank。QLM中不考虑这一项。

# 查询似然模型QLM

- QLM计算公式

$$\begin{aligned} RSV(Q, D) &= P(Q | D) = P(Q | M_D) \\ &= P(q_1 q_2 \dots q_m | M_D) \\ &= P(q_1 | M_D) P(q_2 | M_D) \dots P(q_m | M_D) \\ &= \prod_{w \in Q} P(w | M_D)^{c(w, Q)} \end{aligned}$$

- 于是检索问题转化为估计文档 $D$ 的一元语言模型 $M_D$ ，也即求所有词项 $w$ 的概率 $P(w|M_D)$

# QLM概念理解

- QLM中 $P(Q|D)$ 本质上是 $P(Q|M_D)$ ，不能把 $P(Q|D)$ 称为文档 $D$ （的语言模型 $M_D$ ）生成查询 $Q$ 的概率
- 文档 $D$ 和 $Q$ 都是某个总体分布的样本(实例)，样本(实例)是不会产生样本(实例)的
  - 样本是不会再生成其他东西的，样本只能用来推断总体的某些信息，比如总体的某些未知参数(通过一篇文章来推断作者的风格)
- 同样，不能把 $P(w/M_D)$ 或 $P(w|D)$ 理解为“ $w$ 在文档 $D$ 中的概率”

# QLM求解步骤

- 第一步：根据文档 $D$ (样本)，估计文档模型 $M_D$ (总体)，在一元模型下，即计算所有词项  $w$  的概率  $P(w|M_D)$
- 第二步：计算在模型 $M_D$ 下生成查询 $Q$ 的似然(即概率)
- 第三步：按照得分对所有文档排序

# $M_D$ 的估计

- 问题：已知样本 $D$ ，求其模型 $M_D$ 的参数 $P(w/M_D)$ 。
- 对于该参数估计问题，可以采用最大似然估计 (Maximum Likelihood Estimation, MLE)。
- MLE：使得观察样本出现概率(似然)最大的估计。
  - 一射击世界冠军和一菜鸟打靶，其中一人放一枪得到10环，请问是谁打的？显然世界冠军打的可能性大，也就是说这是使得10环这个事件出现概率最大的估计。

# $M_D$ 的MLE估计

- 设词项词典的大小为 $L$ ，则模型 $M_D$ 的参数可以记为：

$$\begin{aligned}\vec{\theta}_D &= (\theta_1, \theta_2, \dots, \theta_L) \\ &= (P(w_1 | M_D), P(w_2 | M_D), \dots, P(w_L | M_D))\end{aligned}$$

- MLE估计：

$$\vec{\theta}_D^* = \arg \max_{\vec{\theta}_D} P(D | \vec{\theta}_D)$$

- 关键是如何求  $P(D | \vec{\theta}_D)$ ，也就是说假设这些参数已知的情況下，如何求上述概率。

# 总体分布 $M_D$ 的假设

- 两种文本生成模型：
  - 多元贝努利模型(概率模型BIM中使用):  $D$ 是抛 $L$ 个( $L$ 是词项词典的大小)不同的硬币生成的, 每个硬币对应一个词项, 统计所有向上的硬币对应的词项便生成文本 $D$ 。多元贝努利模型中的参数是每个硬币朝上的概率, 共有 $L$ 个。
  - 多项式模型:  $D$ 是抛1个 $L$ 面的骰子抛 $|D|$ 次生成的, 将每次朝上的那面对应的词项集合起来便生成文本 $D$ 。
- QLM在1998年提出时采用的是多元贝努利模型, 后来才有人用多项式模型并发现多项式模型通常优于贝努利模型。所以后来介绍QLM时大都用多项式模型。



# 文本生成的多项式模型

- 有一个 $L$ 个面的不规则骰子，在第 $i$ 个面上写着 $w_i$ ，文档 $D=d_1d_2\dots d_n$ 可以认为是抛 $n$ 次骰子得到的



- 检索过程就是根据观察样本 $D$ 的估计 $Q$ 的生成概率，即在已知抛 $n$ 次的结果为文档 $D$ 的条件下，抛 $m$ 次的结果为查询 $Q$ 的概率 $P(Q|M_D)=?$
- $D = (c(w_1, D), c(w_2, D), \dots, c(w_L, D))$ ， $c(w_i, D)$ 是文档 $D$ 中 $w_i$ 的出现次数
- $D =$  我 喜欢 基于 统计 语言 模型 的 信息 检索 模型
- $D = (<\text{我}, 1>, <\text{喜欢}, 1>, <\text{基于}, 1>, <\text{统计}, 1>, <\text{语言}, 1>, <\text{模型}, 2>, <\text{的}, 1>, <\text{信息}, 1>, <\text{检索}, 1>)$

# 多项随机试验

- 多项(Multinomial)随机试验是二项随机试验(贝努利试验)的扩展，一篇文档 $D$ 可以看成多项随机试验的结果
  - 多项随机试验由 $n$ 次相互独立的子试验组成
  - 每个子试验含有 $L$ 个互斥且完备的可能结果 $w_1, w_2, \dots, w_L$ 。如果 $L=2$ 则是 $n$ 重贝努利试验，对应二项分布 $B(n, p)$ :
$$P(X=k) = C_n^k p^k (1-p)^{n-k}$$
  - 每个子试验中 $w_i$ 发生的概率不变，记为  $\theta_i$

# 多项随机试验(续)

- 设随机变量 $X_1, X_2, \dots, X_L$ 用于记录 $n$ 次子试验中 $w_1, w_2, \dots, w_L$ 的发生次数，实际记录值为 $x_1, x_2, \dots, x_L$ ， $x_1 + x_2 + \dots + x_L = n$ ，如果某个 $w_i$ 不出现，则对应的 $x_i = 0$
- 则该多项随机试验中 $w_1, w_2, \dots, w_L$ 发生次数的联合分布是一个多项式分布：

$$\begin{aligned}
 f(x_1, x_2, \dots, x_L) &= P(X_1 = x_1, X_2 = x_2, \dots, X_L = x_L) = C_n^{x_1} \theta_1^{x_1} C_{n-x_1}^{x_2} \theta_2^{x_2} \dots C_{n-x_1-\dots-x_{L-1}}^{x_L} \theta_L^{x_L} \\
 &= \frac{n!}{x_1!(n-x_1)!} \times \frac{(n-x_1)!}{x_2!(n-x_1-x_2)!} \times \dots \times \frac{(n-x_1-\dots-x_{L-1})!}{x_L!(n-x_1-\dots-x_L)!} \prod_{i=1}^L \theta_i^{x_i} = n! \prod_{i=1}^L \frac{\theta_i^{x_i}}{x_i!}
 \end{aligned}$$

# $M_D$ 的参数求解

■ 求解

$$\vec{\theta}_D^* = \arg \max_{\vec{\theta}_D} P(D | \vec{\theta}_D) = \arg \max_{\vec{\theta}_D} n! \prod_{i=1}^L \frac{\theta_i^{c(w_i, D)}}{c(w_i, D)!}$$

$$\sum_{i=1}^L \theta_i = 1$$

■ 条件极值问题，采用拉格朗日乘数法求解，得到拉格朗日函数：

$$L(\lambda, \vec{\theta}_D) = n! \prod_{i=1}^L \frac{\theta_i^{c(w_i, D)}}{c(w_i, D)!} + \lambda(1 - \sum_{i=1}^L \theta_i)$$

■ 对每个 $\theta_i$ 求偏导，令其为0，解得：

$$\theta_i^* = P_{ML}(w_i | M_D) = \frac{c(w_i, D)}{\sum_{j=1}^L c(w_j, D)} = \frac{c(w_i, D)}{|D|}$$

# 一个MLE估计的例子

- $D = (<\text{我}, 1>, <\text{喜欢}, 1>, <\text{基于}, 1>, <\text{统计}, 1>, <\text{语言}, 1>, <\text{模型}, 2>, <\text{的}, 1>, <\text{信息}, 1>, <\text{检索}, 1>)$

- 采用MLE估计有：

$$\begin{aligned} P(\text{我}|M_D) &= P(\text{喜欢}|M_D) = P(\text{基于}|M_D) \\ &= P(\text{统计}|M_D) = P(\text{语言}|M_D) = P(\text{的}|M_D) \\ &= P(\text{信息}|M_D) = P(\text{检索}|M_D) = 0.1 \end{aligned}$$

$$P(\text{模型}|M_D) = 0.2$$

其他词项的概率为0

# MLE估计的零概率问题

---

- 对于任意不属于 $D$ 的词汇，其概率的MLE估计值为0(数据的稀疏性)。然而，样本 $D$ 中不出现的词，并不代表在新文本中不出现。
- 类比：作者写的一篇文章里面不用某个词，并不代表以后不用这个词

# MLE估计零概率的一个例子

- 一个不规则的骰子，分别用1-6之间的6个数字代表每个面。假设连续抛10次后，观测到的结果序列为2132461232(一篇文档)。问：抛3次的结果序列为325(一个查询)的概率是多大？

# 例子(续)

- 其实就是求：在已知抛10次的观测结果为2132461232的条件下，抛3次的观测结果为325的概率 $P(325 | 2132461232)$ .
- 用 $p_i$ 表示第 $i$ 个面朝上的概率，则 $p_i$ 是一个常数
- 若已知 $p_i (1 \leq i \leq 6)$ ，则显然抛3次的观测结果为325的概率为 $p_3 p_2 p_5$
- 但是每个 $p_i$ 未知，已知的只是抛10次的观测结果2132461232。
- 需要由观测结果2132461232去估计各个 $p_i (1 \leq i \leq 6)$ 。



# 例子中的MLE估计

- 抛骰子的例子中：  $P(325|2132461232) = ?$  —————→ 估计  $p_i, i=1,2,\dots,6$
- 使用最大似然估计，根据样本2132461232估计  $p_i$   
 $p_1=2/10, p_2=4/10, p_3=2/10, p_4=1/10, p_5=0/10, p_6=1/10$
- 显然，5没有出现在样本中，不能说明该骰子第5面永远不会朝上。  
更严重的是，  $P(325|2132461232) = p_3 p_2 p_5 = 0!$
- 因此，上述的估计结果需要调整，使所有  $p_i > 0$ 。
- 思想：从  $p_1, p_2, p_3, p_4$  的估计值中扣出一点点给  $p_5$
- 最简单的方法：Add-One  
 $p_1=3/16, p_2=5/16, p_3=3/16, p_4=2/16, p_5=1/16, p_6=2/16$
- 从上述例子总结出以下几点：
  - (1) 因样本的数据稀疏性，最大似然估计(MLE)导致零概率问题
  - (2) 必须设法调整MLE使得所有事件的概率都大于0 → 平滑(smoothing)

# 数据平滑的一般形式

Discounted(折扣后的) Maximum Likelihood

$$p(w|D) = \begin{cases} p_{DML}(w|D) & w \in D \\ \alpha_D p(w|REF) & otherwise \end{cases}$$

Reference Language Model

$$\alpha_D = \frac{1 - \sum_{w \in D} p_{DML}(w|D)}{\sum_{w \notin D} p(w|REF)}$$

Collection Language Model

$$p(w|REF) = \underline{p(w|C)} = \frac{\sum_D c(w, D)}{\sum_w \sum_D c(w, D)}$$

- 在IR中，一般取
- $p(w/C)$  等于  $w$  出现的次数除以所有词出现的次数。

# 几种QLM中常用的平滑方法

- Jelinek-Mercer(JM),  $0 \leq \lambda \leq 1$ , 文档模型和文档集模型的混合

$$p(w|D) = \lambda p_{ML}(w|D) + (1 - \lambda)p(w|C)$$

- 课堂提问, 对于  $w \in D$ , 折扣后的  $P_{DML}(w|D)$  是不是一定小于  $P_{ML}(w/D)$ ?

- Dirichlet Priors(Dir),  $\mu \geq 0$ , *DIR*和*JM*可以互相转化

$$p(w|D) = \frac{c(w,D) + \mu p(w|C)}{|D| + \mu}$$

- Absolute Discounting(Abs),  $0 \leq \delta \leq 1$ ,  $|D|_u$ 表示 $D$ 中不相同的词个数( $u$ =unique)

$$p(w|D) = \frac{\max(c(w,D) - \delta, 0)}{|D|} + \frac{\delta |D|_u}{|D|} p(w|C)$$

# QLM的求解过程图示

	$w_1$	$w_2$	$w_3$	...	$w_L$
$D_1$	$P_{11} \checkmark$	$P_{12}$	$P_{13} \checkmark$	...	$P_{1L}$
$D_2$	$P_{21} \checkmark$	$P_{22} \checkmark$	$P_{23}$	...	$P_{2L}$
...	...	...	...	...	...
$D_N$	$P_{N1}$	$P_{N2} \checkmark$	$P_{N3} \checkmark$	...	$P_{NL}$

先计算 $P_{ML}$ ，然后采用平滑公式计算 $P_{ML} \rightarrow P_{DML}$

# 几种QLM中常用的平滑方法

- Jelinek-Mercer(JM),  $0 \leq \lambda \leq 1$ , 文档模型和文档集模型的混合

$$p(w|D) = \lambda p_{ML}(w|D) + (1 - \lambda)p(w|C)$$

- 课堂提问, 对于  $w \in D$ , 折扣后的  $P_{DML}(w|D)$  是不是一定小于  $P_{ML}(w/D)$ ?

- Dirichlet Priors(Dir),  $\mu \geq 0$ , *DIR*和*JM*可以互相转化

$$p(w|D) = \frac{c(w,D) + \mu p(w|C)}{|D| + \mu}$$

- Absolute Discounting(Abs),  $0 \leq \delta \leq 1$ ,  $|D|_u$ 表示 $D$ 中不相同的词个数( $u$ =unique)

$$p(w|D) = \frac{\max(c(w,D) - \delta, 0)}{|D|} + \frac{\delta |D|_u}{|D|} p(w|C)$$

# 文档排名函数的转换

■ 
$$P(Q | D) = \prod_{w \in Q} p(w | D)^{c(w, Q)} \quad p(w | D) = \begin{cases} p_s(w | D) & w \in D \\ p_u(w | D) & \text{otherwise} \end{cases}$$

$$\begin{aligned} \log P(Q | D) &= \sum_{w \in Q} c(w, Q) \log p(w | D) = \sum_w c(w, Q) \log p(w | D) \\ &= \sum_{w \in D} c(w, Q) \log p_s(w | D) + \sum_{w \notin D} c(w, Q) \log p_u(w | D) \\ &= \sum_{w \in D} c(w, Q) \log p_s(w | D) + \sum_w c(w, Q) \log p_u(w | D) - \sum_{w \in D} c(w, Q) \log p_u(w | D) \\ &= \sum_{w \in D \cap Q} c(w, Q) \log \frac{p_s(w | D)}{p_u(w | D)} + \sum_{w \in Q} c(w, Q) \log p_u(w | D) \end{aligned}$$

■ 将  $p_s(w | D) = p_{DML}(w | D), p_u(w | D) = \alpha_D p(w | C)$  代入

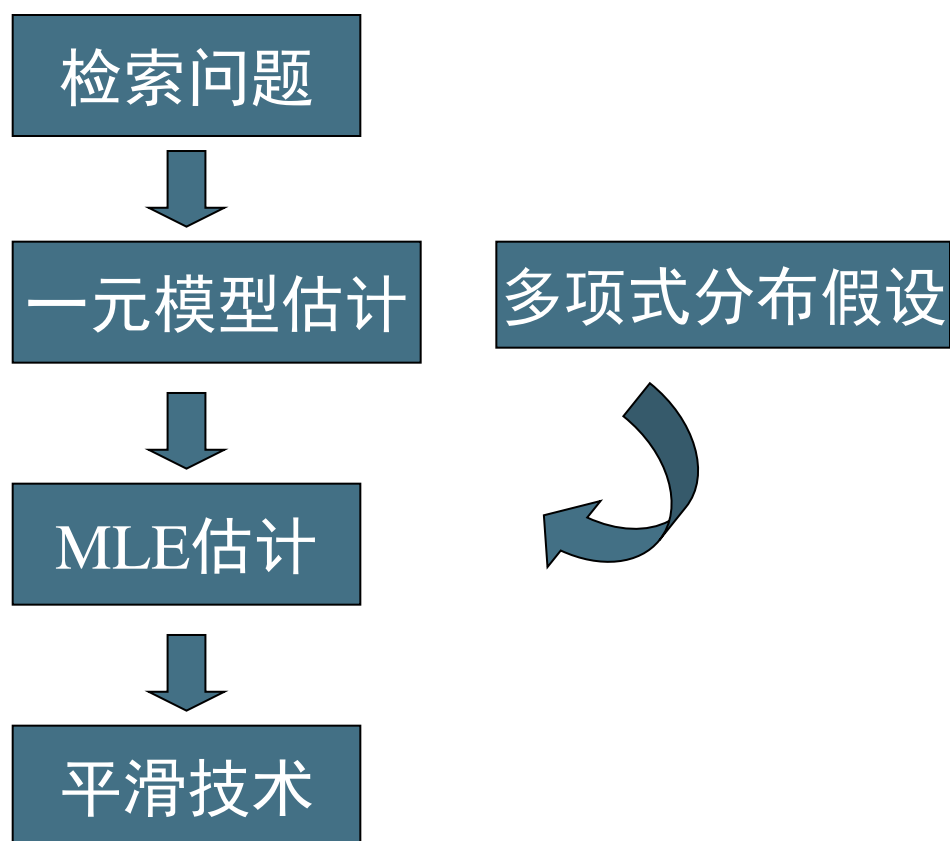
$$\log P(Q | D) = \sum_{w \in Q \cap D} c(w, Q) \log \frac{p_s(w | D)}{\alpha_D p(w | C)} + \boxed{|Q| \log \alpha_D} + \boxed{\sum_{w \in Q} c(w, Q) \log p(w | C)}$$

查询中w的总次数      不影响排名

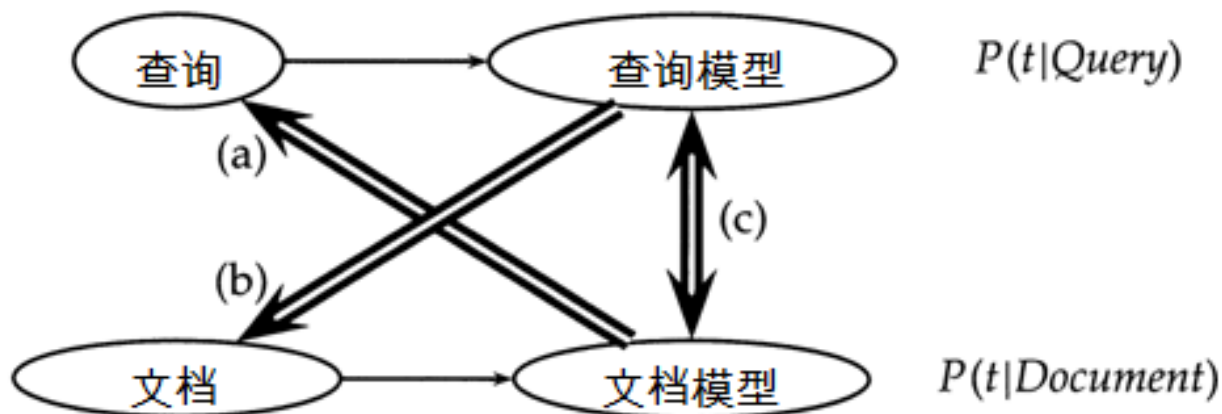
■ 最终排名函数:  $RSV(Q, D) = \sum_{w \in Q \cap D} \boxed{c(w, Q)} \log \frac{\boxed{p_{DML}(w | D)}}{\alpha_D \boxed{p(w | C)}} + |Q| \log \boxed{\alpha_D}$

TF      DF      长度

# QLM模型小结



# 基本SLMIR模型的扩展



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

- 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
- 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
- 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

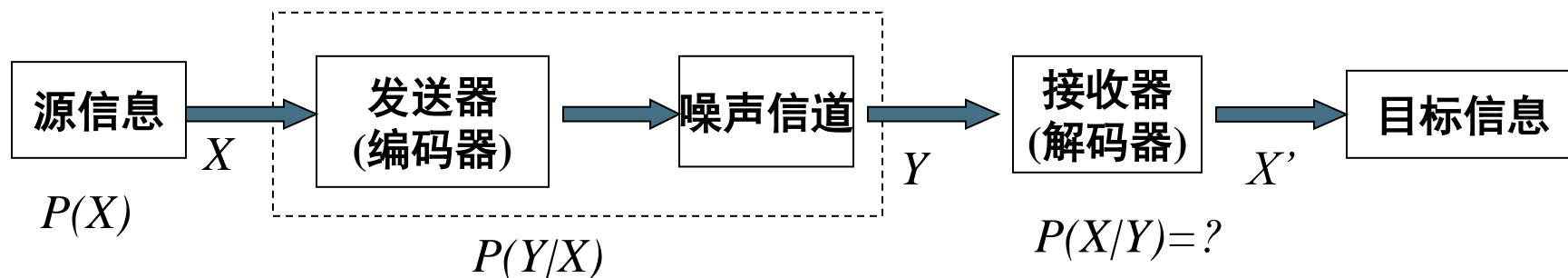


# 其它SLMIR模型

---

- 翻译模型(Translation Model)
- KL距离模型(KL Divergence Model)

# 香农(Shannon)信道



$$\hat{X} = \arg \max_X p(X | Y) = \arg \max_X p(Y | X) p(X)$$

当 $X$ 是文本时, $p(X)$ 就是一个语言模型


一些例子:

语音识别:	$X$ =词序列	$Y$ =语音信号
机器翻译(中->英):	$X$ =英文句子	$Y$ =中文句子
OCR:	$X$ =纠错后的词	$Y$ =错误的词语
文档摘要:	$X$ =摘要	$Y$ =文档
信息检索:	$X$ =文档	$Y$ =查询

# 基于翻译模型的IR模型

- 基本的QLM模型不能解决词语失配(word mismatch)问题, 即查询中的用词和文档中的用词不一致, 如: 电脑 vs. 计算机
- 假设 $Q$ 通过一个有噪声的香农信道变成 $D$ , 从 $D$ 估计原始的 $Q$

翻译概率                      生成概率



$$P(Q | D) = \prod_i P(q_i | D) = \prod_i \sum_j P(q_i | w_j) P(w_j | M_D)$$

- 翻译概率 $P(q_i | w_j)$ 在计算时可以将词项之间的关系融入。
  - 基于词典来计算(人工或者自动构造的同义词/近义词/翻译词典)
  - 基于语料库来计算(标题、摘要 vs. 文本; 文档锚文本 vs. 文档)

# KL距离(相对熵)模型

$$Score(Q, D) = \log \frac{P(Q | M_D)}{P(Q | M_C)}$$

$$Score(Q, D) = \sum_{q_i \in Q} tf(q_i, Q) * \log \frac{P(q_i | M_D)}{P(q_i | M_C)}$$

$$\propto \sum_{q_i \in Q} P(q_i | M_Q) * \log \frac{P(q_i | M_D)}{P(q_i | M_C)}$$

$$= \sum_{q_i \in Q} P(q_i | M_Q) * \log \frac{P(q_i | M_D)}{P(q_i | M_Q)} - \sum_{q_i \in Q} P(q_i | M_Q) * \log \frac{P(q_i | M_C)}{P(q_i | M_Q)}$$

$$= -KL(M_Q, M_D) + \boxed{KL(M_Q, M_C)}$$

$$\propto -KL(M_Q, M_D) = \sum_{q_i \in Q} P(q_i | M_Q) * \log P(q_i | M_D) - \boxed{\sum_{q_i \in Q} P(q_i | M_Q) * \log P(q_i | M_Q)}$$

$$\propto \sum_{q_i \in Q} \boxed{P(q_i | M_Q) * \log P(q_i | M_D)}$$

负的交叉熵

$$P(Q | M_D) = \frac{|Q|!}{\prod_{q_i \in Q} tf(q_i, Q)!} \prod_{q_i \in Q} P(q_i | D)^{tf(q_i, Q)}$$

$$P(Q | M_C) = \frac{|Q|!}{\prod_{q_i \in Q} tf(q_i, Q)!} \prod_{q_i \in Q} P(q_i | C)^{tf(q_i, Q)}$$

多项分布

对同一-Q, 为常数

# 统计语言建模IR模型优缺点

---

- 优点：
  - 理论上具有解释性，有扩展空间
  - 有些模型虽然计算上仍然依赖于term独立性假设，但是模型本身并不依赖于term独立性假设。
- 缺点：
  - 数据稀疏性，需要参数估计

# 提纲

- ① 上一讲回顾
- ② 统计语言模型 (SLM)
- ③ 基于SLM的IR模型
- ④ SLMIR模型讨论

# SLMIR vs. VSM (1)

- SLMIR中有一些东西和VSM一样
- 词项频率直接在模型中使用
  - 但是在SLMIR中没有进行放缩变化(scaled)
- 本质上概率表示已经进行了长度归一化
  - VSM中的余弦归一化也做了类似工作
- 文档中的词项频率和文档集频率混合以后和idf的效果相当
  - 那些文档集中比较罕见，但是某些文档中比较普遍的词项将对排序起更重要的影响。

# SLMIR vs. VSM (2)

- SLMIR vs. VSM：共性
  - 模型中都直接使用了词项频率
  - 本质上概率表示已经进行了长度归一化
  - 文档中词项频率和文档集频率混合以后和idf的效果相当
- SLMIR vs. VSM：不同
  - SLMIR：基于概率论
  - VSM：基于相似度，一个线性代数中的概念
  - 文档集频率 vs. 文档概率
  - 词项频率、归一化等计算细节



# 基于统计建模的IR模型: 假设

- 简化假设：查询和文档是同一类对象，与实际并不相符！
  - 已经出现了一些不采用上述假设的SLMIR模型
  - VSM也基于同一假设
- 简化假设：词项之间是独立的
  - 同样，VSM中也采用了词项独立性假设
- 比向量空间中的假设表述更清晰
- 因此，比VSM具有更好的理论基础
  - ... 但是纯语言模型的效果会大大低于经过精心调参的向量模型的效果。

# 参考资料

---

- 《信息检索导论》第12章
- Ponte and Croft's 1998 SIGIR paper (one of the first on LMs in IR)
- Lemur toolkit (能够很好地支持语言模型)
- ChengXiang Zhai. Statistical Language Models For Information Retrieval, 一本综述书籍