



# 基于统计的中文语言建模

---

刘秉权

智能技术与自然语言处理研究室

新技术楼612室

Email: [liubq@hit.edu.cn](mailto:liubq@hit.edu.cn)



# 主要内容

---

- 定义
- 构造方法
- 数据稀疏
- 评价标准
- 主要统计语言模型
- 统计语言模型的作用



# 统计语言模型(Statistical Language Model)

- Statistical Language Model
- 统计语言模型试图捕获自然语言的统计规律以改善各种自然语言应用系统的性能
- 广泛地应用于语音识别、手写体文字识别、机器翻译、键盘输入、信息检索等领域
- 统计语言建模(Statistical Language Modeling)相当于对各种语言单位如字、词、句子或整篇文章进行概率分布的估计



# 定义

---

给定所有可能的句子  $s$ （还可以是其他语言单位），统计语言模型就是一个概率分布  $p(s)$ 。

迄今为止，几乎所有的语言模型将一个句子的概率分解为条件概率的乘积：

$$p(s) = p(w_1, \dots, w_n) = \prod_{i=1}^n p(w_i | h_i)$$

这里  $w_i$  是句中的第  $i$  个词， $h_i = \{w_1, w_2, \dots, w_{i-1}\}$  称为历史。



# 例子

---

$$\begin{aligned} & p(\text{我是一个学生}) \\ &= p(\text{我, 是, 一, 个, 学生}) \\ &= p(\text{我}) \cdot \\ &\quad p(\text{是} \mid \text{我}) \cdot \\ &\quad p(\text{一} \mid \text{我, 是}) \cdot \\ &\quad p(\text{个} \mid \text{我, 是, 一}) \cdot \\ &\quad p(\text{学生} \mid \text{我, 是, 一, 个}) \end{aligned}$$



# N-gram模型

---

实际应用中，由于严重的数据稀疏和系统处理能力的限制，统计语言建模只能考虑有限长度的历史。

通过将语言模拟成  $N-1$  阶马尔科夫源，N-gram 模型减少了参数估计的维数：

$$p(w_i | h_i) \approx p(w_i | w_{i-N+1}, \dots, w_{i-1})$$

$N$  值的选择要考虑参数估计的稳定性和描述能力的折衷。Trigram 和 Bigram 是通常的选择。



## 例子 (Bigram, Trigram)

$p(\text{我是一个学生})$   
 $= p(\text{我, 是, 一, 个, 学生})$   
 $= p(\text{我}) \cdot$

$p(\text{是} | \text{我}) \cdot$  二元

$p(\text{一} | \text{是}) \cdot$

$p(\text{个} | \text{一}) \cdot$

$p(\text{学生} | \text{个})$

$p(\text{我是一个学生})$   
 $= p(\text{我, 是, 一, 个, 学生})$   
 $= p(\text{我}) \cdot$

$p(\text{是} | \text{我}) \cdot$  三元

$p(\text{一} | \text{我, 是}) \cdot$

$p(\text{个} | \text{是, 一}) \cdot$

$p(\text{学生} | \text{一, 个})$



# SLM的参数学习

## 1. 有指导的参数学习

是一个基于完全数据的极大似然性估计(Maximum Likelihood Estimation, MLE)。

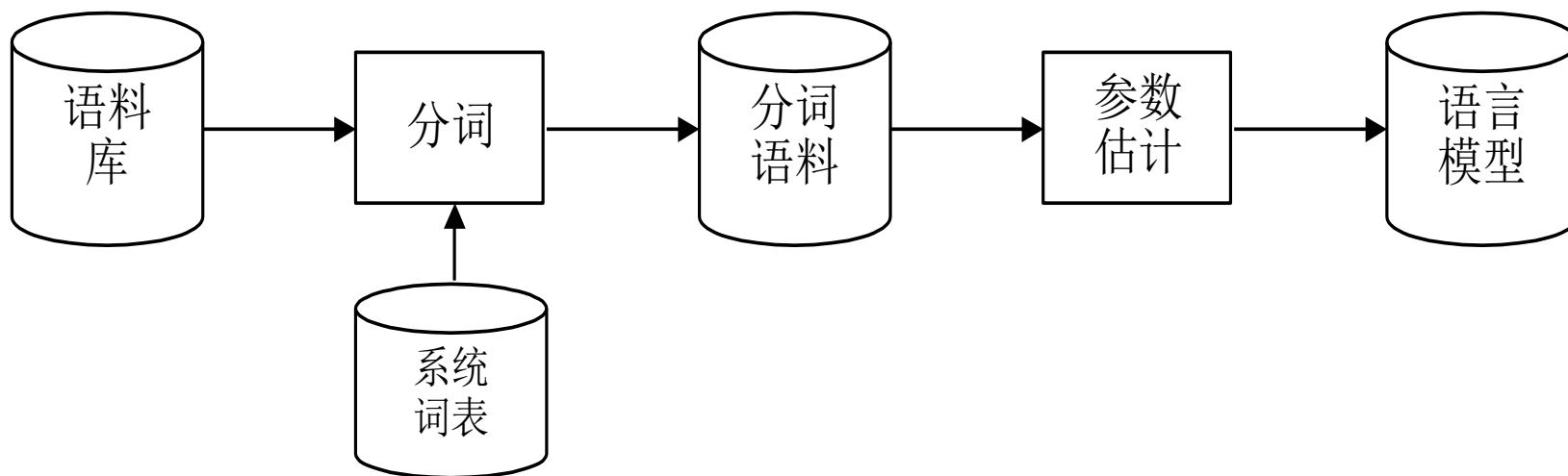
设  $Count(x)$  为模型所预测的一个事件  $x$  在训练语料中出现的次数,  $Count(y)$  为语料中所有入选的相应的条件事件  $y$  的观察数, 则模型所描述的事件  $x$  的概率可以由下式估计:  $p(x) = f(x) = \frac{Count(x)}{Count(y)}$ , 式中,  $f(x)$  为相对频度函数。

## 2. 无指导的参数学习

是一个具有隐含变量的参数训练过程。与有指导的参数学习方法相比, 无指导方法所依赖的训练集可以是不完全数据(incomplete data), 因而不需事先进行人工加工。



# 参数训练系统



# N-gram模型的概率估计

## ■ 极大似然估计

$$p(w_i | w_{i-N+1} \cdots w_{i-1}) = \frac{c(w_{i-N+1} \cdots w_i)}{\sum_{w_i} c(w_{i-N+1} \cdots w_i)}$$

$$= \frac{c(w_{i-N+1} \cdots w_i)}{c(w_{i-N+1} \cdots w_{i-1})}$$

分母为长度为N-1的  
“历史”，分子为长  
度为N的整个句子

$$P(\text{朋友} | \text{漂亮}) = c(\text{漂亮}, \text{朋友}) / c(\text{漂亮})$$



# 数据稀疏

由于语言模型的训练文本 $T$ 的规模及其分布存在着一定的局限性和片面性，许多合理的语言搭配现象没有出现在 $T$ 中。例如：一个词串 $w_{i-N+1} \cdots w_i$ 没有出现在训练文本 $T$ 中，该词串对应的上下文条件概率 $p(w_i | w_{i-N+1} \cdots w_{i-1}) = 0$ ，从而导致该词串所在的语句 $S$ 的出现概率 $p(S) = 0$ 。

这种情况通常被称作数据稀疏问题（Data Sparseness）或零概率问题。



# N-gram模型的数据稀疏

假设模型训练的词表为 $V$ ,采用 $N$ 元模型,  
则理论上的参数空间大小为 $|V|^N$ , 如果  
 $|V| = 20000$ ,  $N = 3$ , 则 $|V|^N$ 的值将达到  
 $8 \times 10^{12}$ , 需要至少上万 $G$ 的空间来存储。而实际应用中, 模型的大小要远远小于这个理想值。

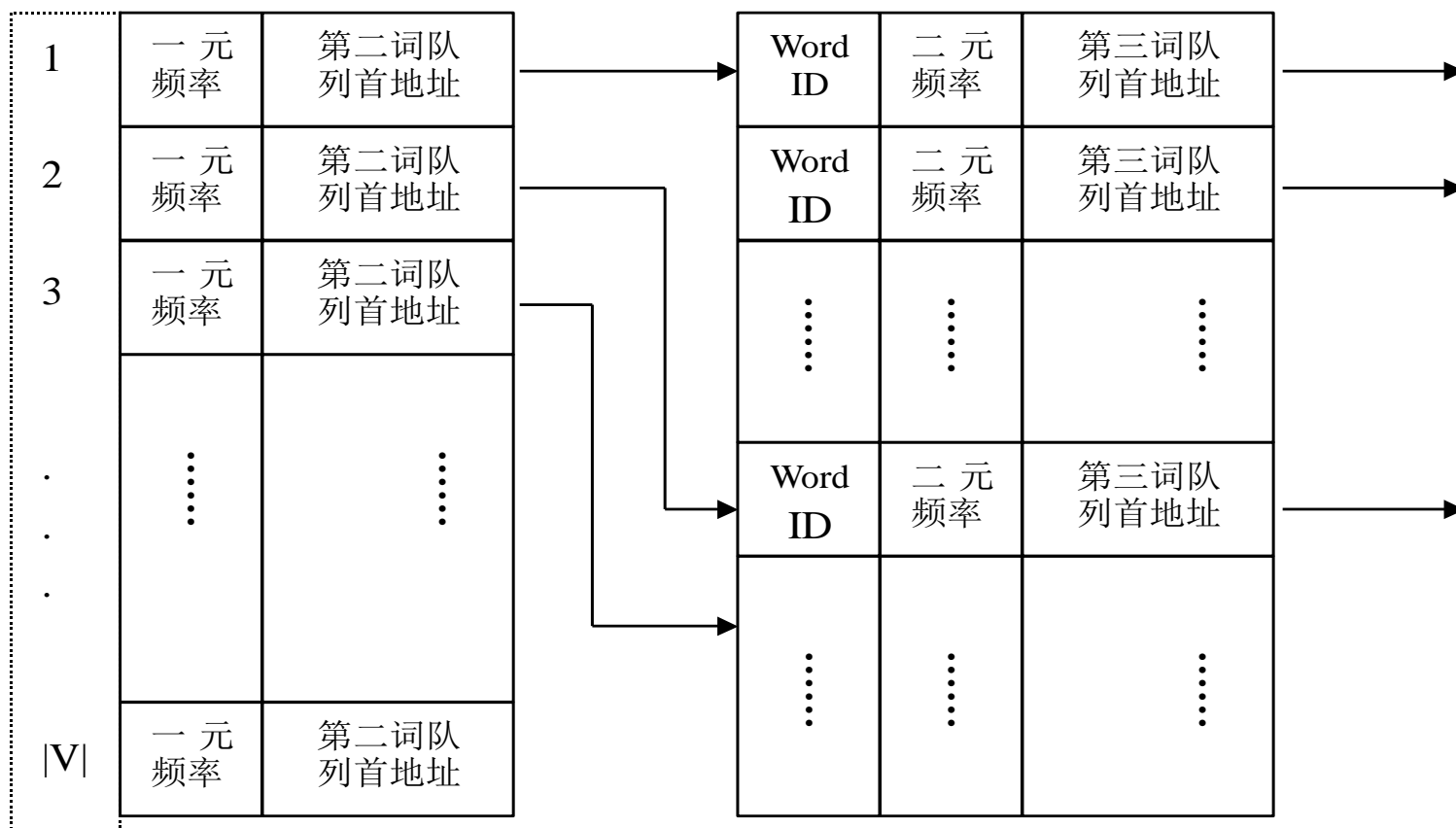


# Zipf统计定律

---

- 对于大量的低频词 (N元对) 来说, 无论训练语料库的规模如何扩大, 其出现频度依然很低或根本不出现, 无法获得其足够的统计特性, 用于可靠的概率估计。
- 直接根据频度对N-gram概率进行极大似然性估计是不可取的

# N-gram语言模型的一种存储结构





# 数据平滑(Smoothing)

- 平滑：对根据极大似然估计原则得到的概率分布进一步调整，确保统计语言模型中的每个概率参数均不为零，同时使概率分布更加趋向合理、均匀
- 数据平滑技术
  - 加法平滑
  - Good-Turing估计
  - 回退平滑(Backing-off Smoothing)
  - 线性插值(Linear Interpolation)
  - 类模型、变长N-gram模型等



## 加法平滑

---

为了避免零概率问题,将 N-gram 模型中每个 N 元对的出现次数加上一个常数  $\delta$  ( $0 < \delta \leq 1$ ), 相应的 N-gram 模型参数  $p_{add}(w_i | w_{i-n+1}^{i-1})$  计算如下:

$$p_{add}(w_i | w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^i) + \delta}{\sum_{w_i} c(w_{i-n+1}^i) + \delta |V|}$$





# Good-Turing估计

对于 N-gram 模型中出现  $r$  次的 N 元对  $w_{i-n+1}^i$ ，根据 Good-Turing 估计公式，

该 N 元对的出现次数为<sup>\*</sup>： $r^* = (r + 1) \frac{n_{r+1}}{n_r}$ ，其中  $n_r$  表示 N-gram 的训练集中实际出现  $r$  次的 N 元对的个数。那么，对于 N-gram 中出现次数为  $r$  的 N 元对

$w_{i-n+1}^i$  的出现概率为：
$$p_{GT}(w_{i-n+1}^i) = \frac{r^*}{\sum_{r=0}^{\infty} r^*} n_{\varphi}$$
 不能为零，本身需要平滑。

Good-Turing 估计公式中缺乏利用低元模型对高元模型进行插值的思想，它通常不单独使用，而作为其他平滑算法中的一个计算工具。

# N-gram模型参数的统计结果 (53.7MB语料, 词典59461个词)

出现次数 $r$	各级 N-gram 模型的出现次数= $r$ 的 N 元对的个数						
	1-gram	2-gram	3-gram	4-gram	5-gram	6-gram	7-gram
1	22490	1342337	3474704	4343780	4411786	4158005	3789897
2	21633	585321	811778	629939	449441	324290	50817
3	20785	367848	377958	229827	131690	79959	25215
4	19971	272266	237503	129336	69213	40707	15460
5	19177	215650	168119	85473	44068	25408	10869
6	18441	179525	129037	62726	31841	18097	8255
7	17762	153201	103248	48466	24230	13790	6605
8	17171	133921	85647	39180	19583	11153	5364
9	16620	118833	72637	32571	16109	9179	4433
10	16147	106975	62718	27638	13664	7732	1230
20	12802	52520	24236	9775	4800	2456	594
30	10957	34059	13840	5487	2771	1269	215
50	8723	19421	6683	2532	1148	496	36
100	6226	8601	2392	796	298	93	1
500	2312	1056	223	65	21	5	0
1000	1314	368	62	18	6	0	0
5000	242	21	2	0	0	0	0
10000	101	4	0	0	0	0	0
20000	42	2	0	0	0	0	0
50000	9	0	0	0	0	0	0
100000	4	0	0	0	0	0	0
200000	3	0	0	0	0	0	0



# 回退平滑

当一个 N 元对  $w_{i-n+1}^i$  的出现次数  $c(w_{i-n+1}^i)$  足够大时,  $p_{ML}(w_i | w_{i-n+1}^{i-1})$  是  $w_{i-n+1}^i$  可靠的概率估计。而当  $c(w_{i-n+1}^i)$  不是足够大时, 采用 Good-Turing 估计对其平滑, 将其部分概率折扣给未出现的 N 元对。当  $c(w_{i-n+1}^i) = 0$  时, 模型回退到低元模型, 按着  $p_{katz}(w_i | w_{i-n+2}^{i-1})$  比例来分配被折扣给未出现的 N 元对的概率:

$$p_{katz}(w_i | w_{i-n+1}^{i-1}) = \begin{cases} p_{ML}(w_i | w_{i-n+1}^{i-1}) & \text{if } (c(w_{i-n+1}^i) \geq k) \\ \alpha \cdot p_{GT}(w_i | w_{i-n+1}^{i-1}) & \text{if } (1 \leq c(w_{i-n+1}^i) < k) \\ \beta \cdot p_{katz}(w_i | w_{i-n+2}^{i-1}) & \text{if } (c(w_{i-n+1}^i) = 0) \end{cases}$$

其中,  $p_{ML}(w_i | w_{i-n+1}^{i-1})$  为最大似然估计模型。  $p_{GT}(w_i | w_{i-n+1}^{i-1})$  为 Good-Turing 概率估计。

阈值  $k$  为一个常量。参数  $\alpha$  和  $\beta$  保证模型参数概率的归一化约束条件, 即

$$\sum_{w_i} p_{katz}(w_i | w_{i-n+1}^{i-1}) = 1。$$



# 线性插值平滑

利用低元 N-gram 模型对高元 N-gram 模型进行线性插值。

平滑公式：

$$p_{\text{interp}}(w_i | w_{i-n+1}^{i-1}) = \lambda_{w_{i-n+1}^{i-1}} \cdot p_{ML}(w_i | w_{i-n+1}^{i-1}) + (1 - \lambda_{w_{i-n+1}^{i-1}}) \cdot p_{\text{interp}}(w_i | w_{i-n+2}^{i-1})$$

N-gram 模型可以递归地定义为由最大似然估计原则得到的 N-gram 模型和(N-1)-gram 模型的线性插值。为了结束以上的递归定义：可以令 Unigram 模型为最大似然估计模型，或者令 0-gram 模型为一个均匀分布模型

$$p_{\text{0阶}}(w_i) = \frac{1}{|V|}。$$

插值系数  $\lambda_{w_{i-n+1}^{i-1}}$ ，可以采用 Baum-Welch 算法估计出来，也可根据经验人工给出。



# 线性插值平滑的简单形式

以 Tri-gram 模型为例：

$$p_{\text{interp}}(w_i | w_{i-2}w_{i-1}) = \alpha \cdot p_{ML}(w_i | w_{i-2}w_{i-1}) + \beta \cdot p_{ML}(w_i | w_{i-1}) + \gamma \cdot p_{ML}(w_i)$$

其中：  $1 \geq \alpha, \beta, \gamma \geq 0, \alpha + \beta + \gamma = 1$



# 基于词类的N-gram模型

设  $C_i$  为词  $w_i$  所属的类，多种基于类的模型结构可被使用。典型地，一个Trigram可选择如下计算方法：

$$p(w_3 | w_1, w_2) = p(w_3 | C_3) \cdot p(C_3 | C_1, C_2)$$





# 类模型提出的意义

---

- 降低模型参数的规模
- 数据稀疏问题的一种解决方式



# 构造方法

---

- 采用语言学家构造的词的语法分类体系，按词性（Part-of-Speech）进行词类划分，借助于词性标注技术，构造基于词性的N-POS模型
- 采用词的自动聚类技术，自动构造基于词的自动聚类的类N-gram模型





# 几种模型的比较

- 基于词的N-gram模型对近邻的语言约束关系的描述能力最强，应用程度最为广泛。一般 $N \leq 3$ ，难以描述长距离的语言约束关系
- N-POS模型的参数空间最小，一般不存在数据稀疏问题，可以构造高元模型，用于描述长距离的语言约束关系。但由于词性数目过少，过于泛化，因此又限制了语言模型的描述能力
- 自动聚类生成的词类数量介于词和词性的数量之间，由此建立的类N-gram模型，既不存在严重的数据稀疏问题，又不存在过于泛化问题



# 统计语言模型的评价标准

- 模型评价标准是模型选择或优化时的目标函数
- 评价一个语言模型的优劣最终要看模型对实际应用系统性能的影响，但实际中很难采用这种评价标准
  - 得到应用系统的一个可靠的测试需要处理大规模的数据，十分费时，测试结果易受很多非线性因素的影响
  - 一般不可能找到一个描述系统性能指标和语言模型参数值间关系的方法。而且不同的测试过程和结论也缺乏横向比较的标准
- 语言模型的不确定性可以用信息熵来衡量，模型的不确定性越大，正确估计语言现象的可能性就越小
- 一般不直接采用基于应用系统性能评价的标准，而采用信息熵(entropy)或模型复杂度(perplexity)概念



# 熵(Entropy)

---

熵表示信息源  $X$  每发一个符号所提供的平均信息量。

熵也可以被视为描述一个随机变量的不确定性的数量。一个随机变量的熵越大，它的不确定性越大，正确估计其值的可能性就越小。越不确定的随机变量越需要大的信息量用以确定其值。



# 熵的定义

---

如果  $X$  是一个离散型随机变量，其概率分布为：

$p(x) = P(X = x)$ ， $x \in X$ ，则  $X$  的熵  $H(X)$  为：

$$H(X) = - \sum_{x \in X} p(x) \log_2 p(x), \text{ 其中约定 } 0 \log 0 = 0。$$

$H(X)$  也可以写为  $H(p)$ ，通常熵的单位为二进制位比特（bit）。



# 语言的熵

---

语言的熵反映语言中每个字或词的平均信息量。对语言  $L$ ，用  $x_1^n = x_1, x_2, \dots, x_n$  表示  $L$  中长为  $n$  的字串或词串， $p(x_1^n)$  是  $x_1^n$  的概率。根据信息论原理，若语言  $L$  是各态遍历的、平稳的随机过程，则熵由下式计算：

$$H(L) = -\lim_{n \rightarrow \infty} \frac{1}{n} \log p(x_1^n)$$



# 语言的熵

---

通常分布  $p(x_1^n)$  是未知的，熵的计算是对某种语言模型进行的：

$$H(L) = -\sum_x p(x) \cdot \log p(x)$$



# 交叉熵 (Cross Entropy)

通常使用交叉熵来衡量一个模型的质量，以此评估语言模型 $p_M$

和真实文本（测试文本）分布 $p_T$  的相似程度：

$$H'(p_T; p_M) = - \sum_x p_T(x) \cdot \log p_M(x)$$

对 N-gram 语言模型 $p$ ，模型的交叉熵可由下式计算：

$$H'(L, p) \approx - \frac{1}{LL - N + 1} \sum_{n=N}^{LL} \log p(x_n | x_{n-N+1}^{n-1})$$

其中 $LL$  表示训练语料的长度。



# 复杂度(迷惑度, Perplexity)

复杂度是熵的变形。语言模型  $P_M$  的复杂度定义为

$$PP_M(T) = 2^{H'(P_T; P_M)}$$

模型的复杂度从信息论角度描述一个语言分支数的几何平均，即一个语言表示在平均意义上的可能的分析。直观地，利用一个语言模型预测文本，给定一段历史，当前字或词平均只可能有  $PP_M$  种选择。复杂度越大，模型的语言约束能力越小。因此，一个约束能力强的语言学模型一般应具有较低的复杂度。这样语言模型研究的任务就是寻找复杂度最小的模型。





# 其他语言模型

---

- 各种变长、远距离N-gram模型
- 决策树模型
- 链文法模型
- 最大熵模型
- 整句模型
- 动态自适应模型



# 动态自适应模型

---

## 1. 基于缓存(Cache)的自适应技术

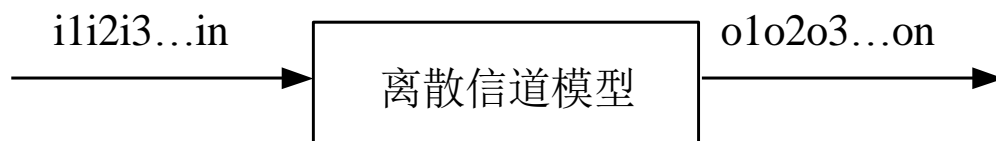
$$p_{adaptive}(w | h) = \lambda p_{static}(w | h) + (1 - \lambda) p_{cache}(w | h)$$

## 2. 主题适应的自适应技术

$$p_{mix}(w_1 w_2 \cdots w_T) = \prod_{i=1}^T \sum_{k=0}^m \lambda_k p_k(w_i | w_{i-N+1} \cdots w_{i-1})$$

# 统计语言模型的作用

- 信源—信道模型：



$$\hat{I} = \arg \max_I (p(I | O)) = \arg \max_I \frac{p(I)p(O|I)}{p(O)} = \arg \max_I p(I)p(O|I)$$

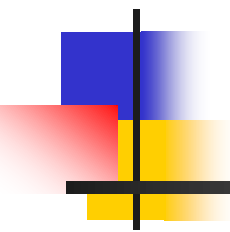
- **I**: 语言文本； **O**: 声音信号、字符图像信号、拼音输入等
- 语言模型：  $p(I)$



# 统计语言模型的不足之处

---

- 统计语言建模技术很少使用真实的语言知识
- 跨领域的适应能力差
- 不能很好地处理长距离语言约束
- 与西方语言相比，汉语的结构更复杂，并且存在独特的分词和生词处理问题



# 谢谢！

---