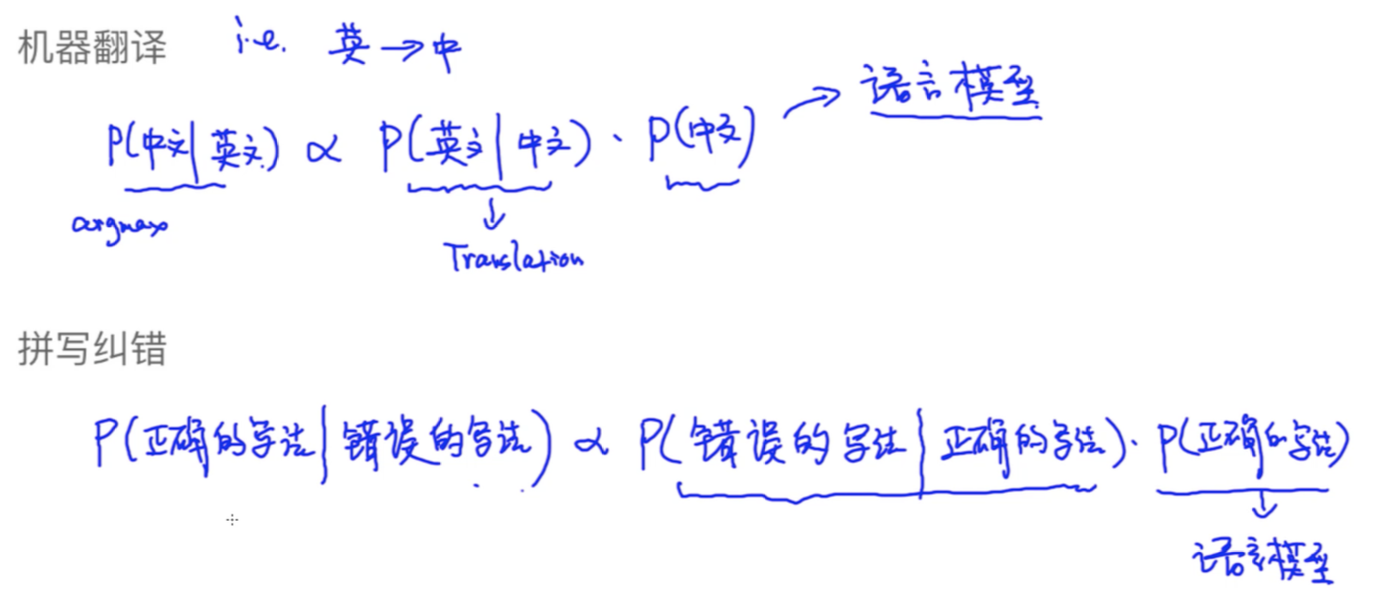
**1. Noisy Channel Model**

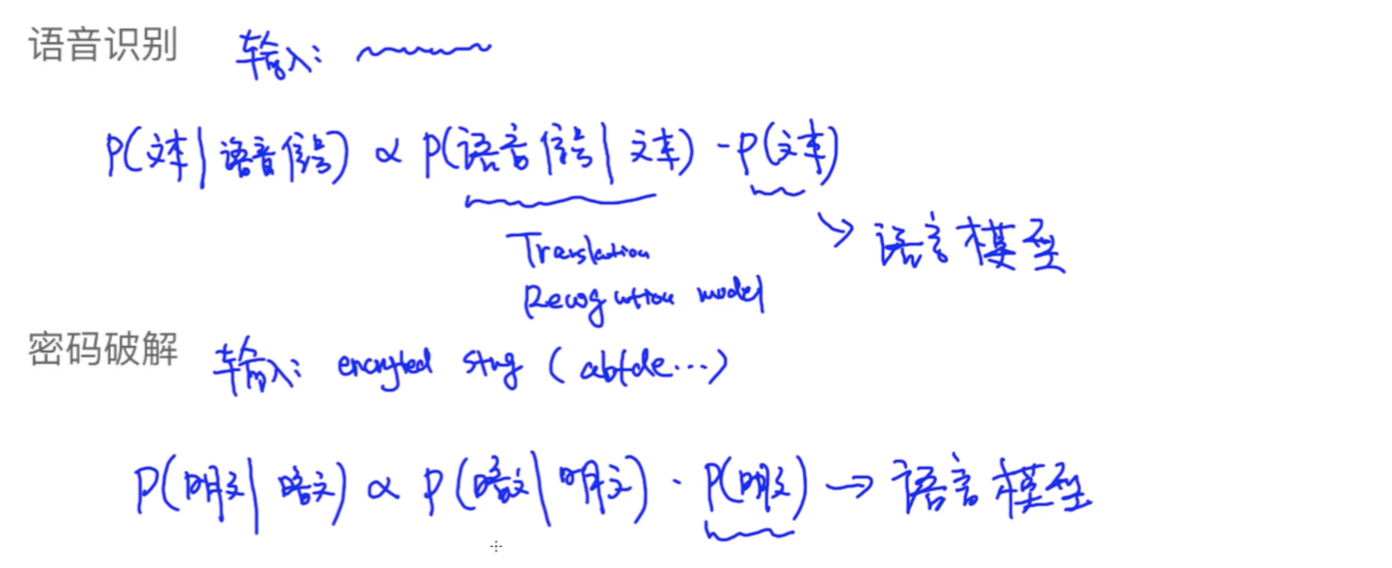


应用场景:

语音识别，机器翻译，拼写纠错，OCR，密码破解

给定一个信号，需要将其转换成文本。

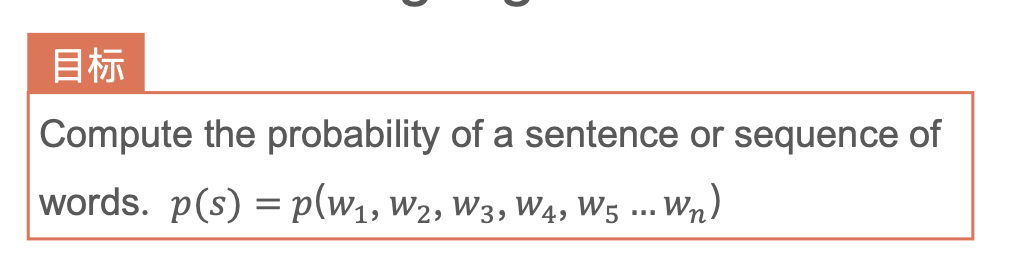




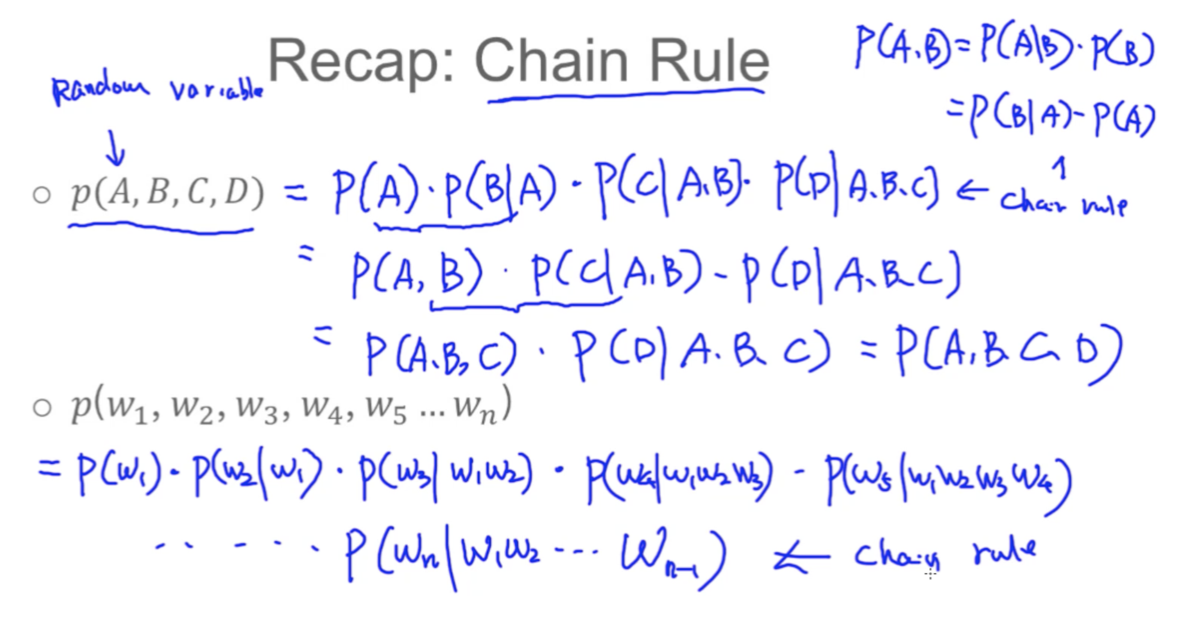
**2. Language Model (LM)**

语言模型用来判断:是否一句话从语法上通顺

LM一般是预先训练好的，pre-trained



**2.1 Chain Rule**



图片包含 物体

描述已自动生成

**2.2 Chain Rule for Language Model**

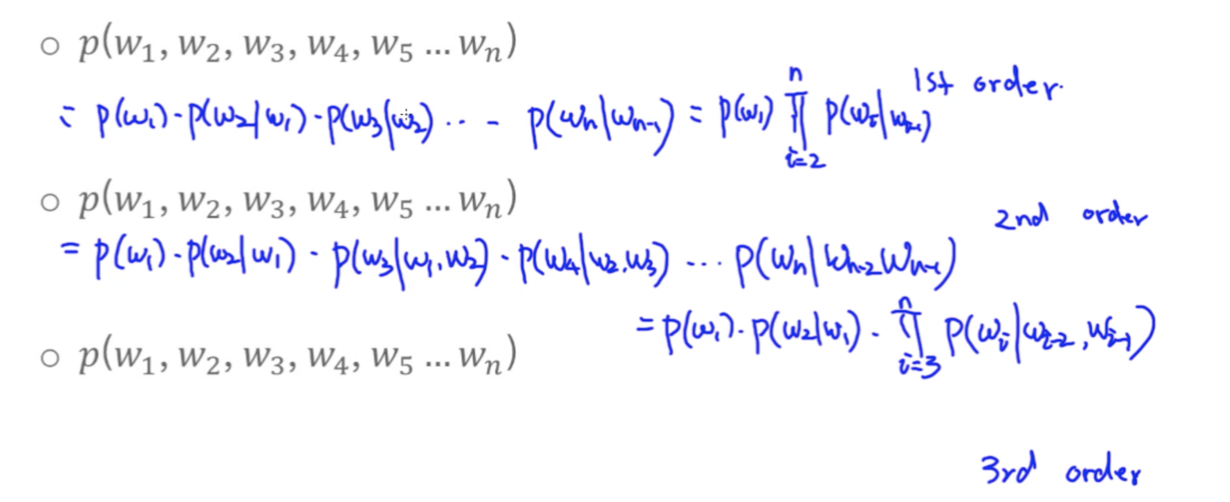
图片包含 文字

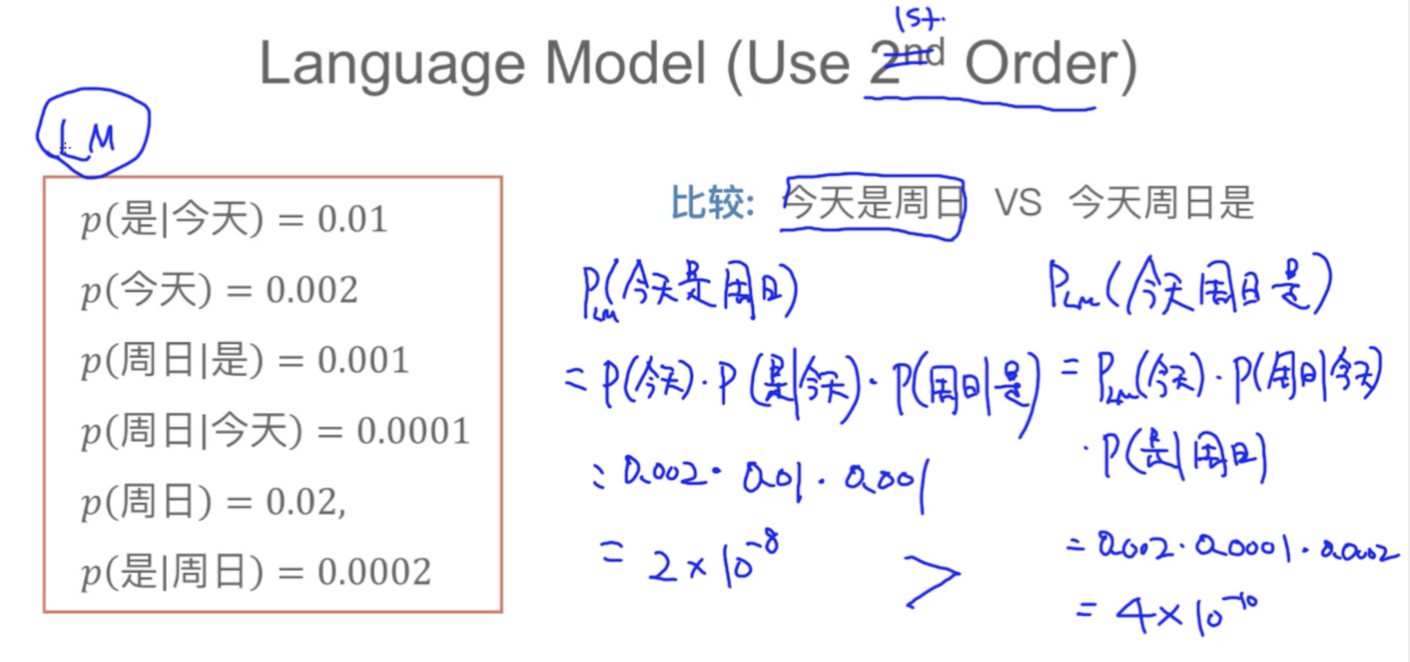
描述已自动生成

当条件中存在多个单词的时候，容易出现“稀疏性”的问题。

**2.3 Markov Assumption**







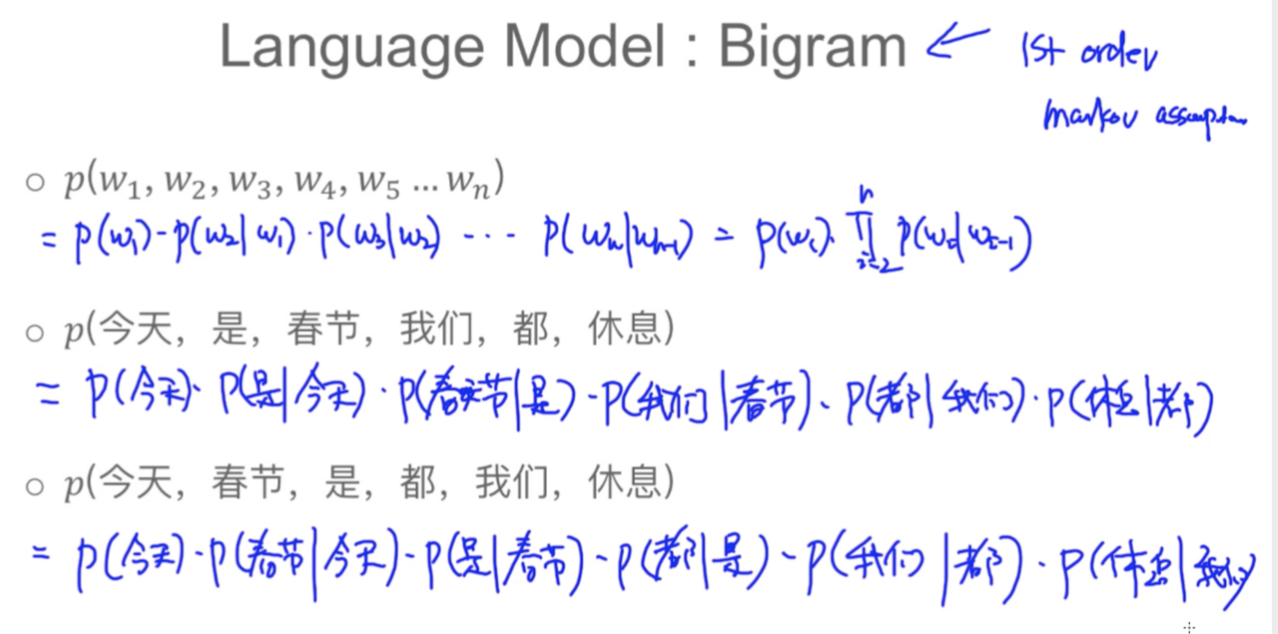
**2.4 Classification**

1）unigram

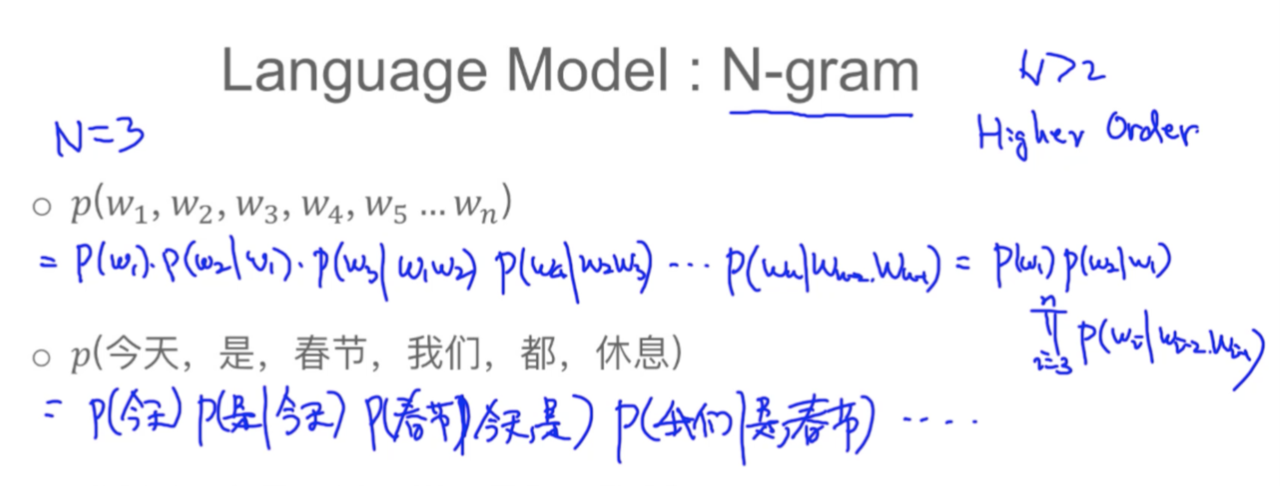
图片包含 文字

描述已自动生成

2）bigram



3）…N-gram



**3. Estimating Probability**

问题：如何去构造这样的语言模型

等效于一个统计问题

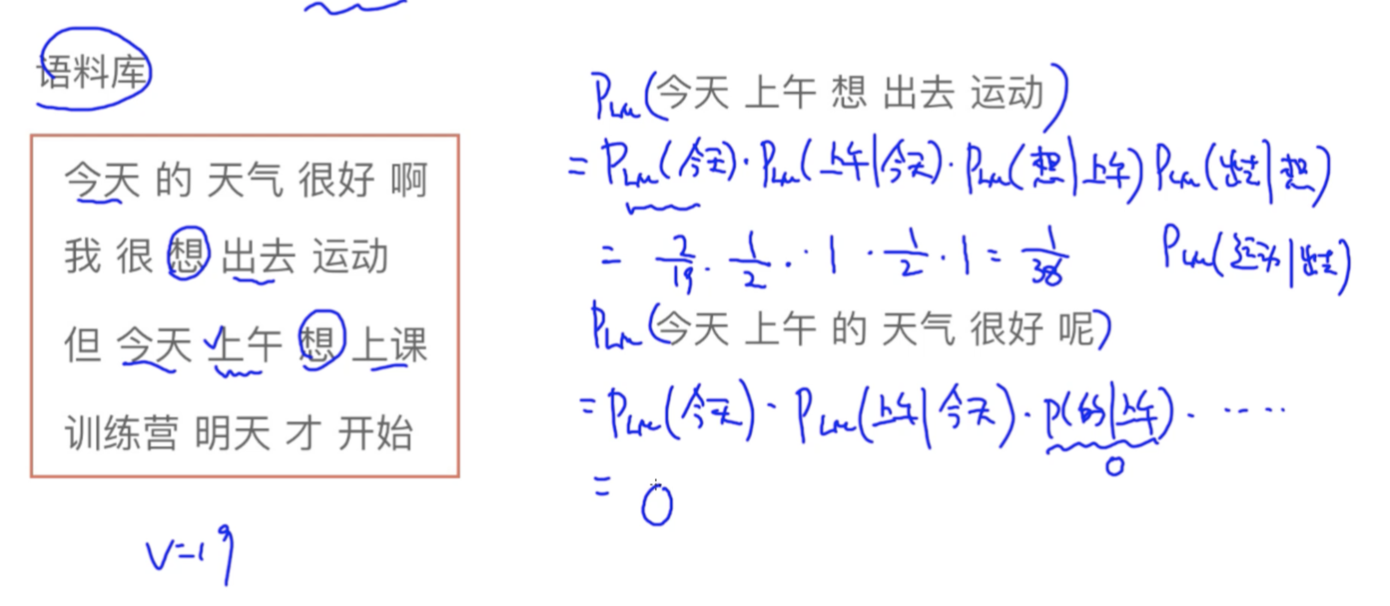
1）unigram

只需要计算每个单词在文章中出现的概率。

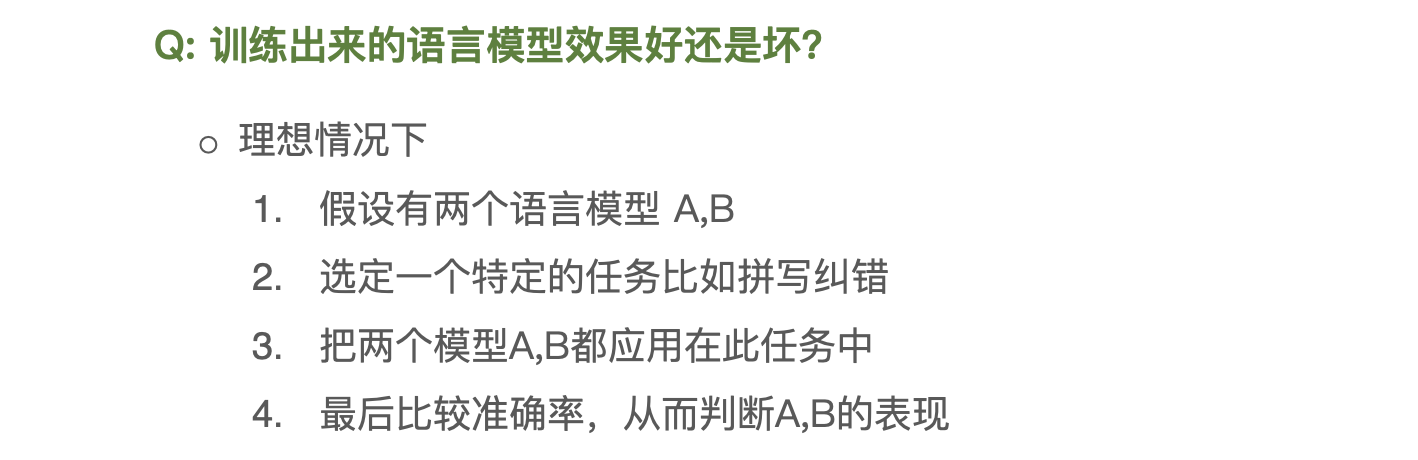
2）bigram

图片包含 人员, 文字

描述已自动生成



**4. Evaluation of Language Model**



缺点：耗时。成本高。依赖于某一个特定任务。

核心思路：从前一个单词推测，最有可能出现的下个单词。（像填空）



**4.1 Perplexity**

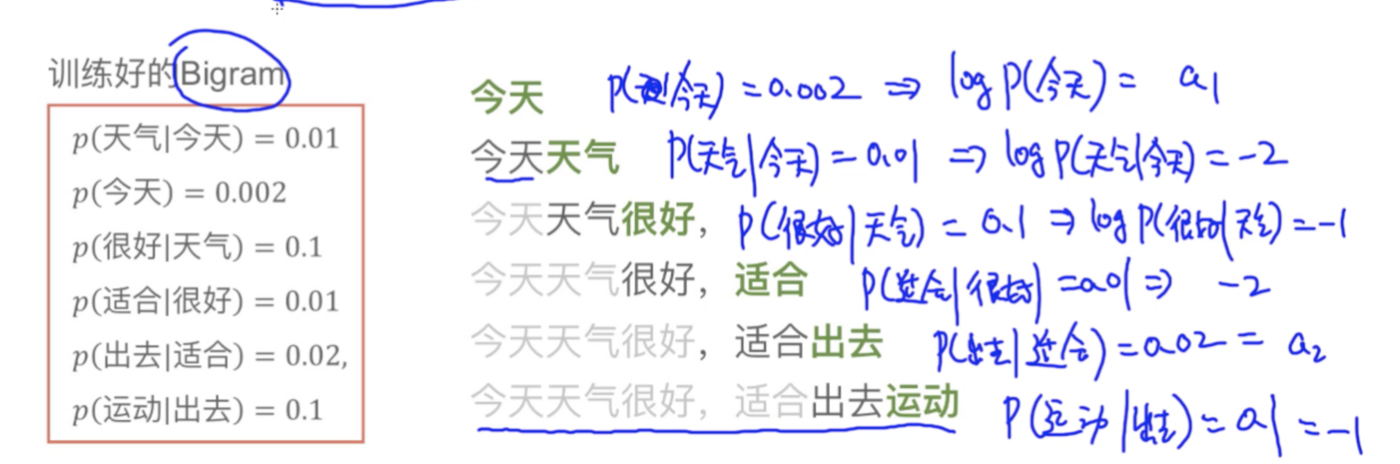
尤其是在无监督学习下，通常使用perplexity来评估。



X通常是，在一个语料库中，训练好的。

一个好的LM，放在一个语料库中，我们希望是越大越好。

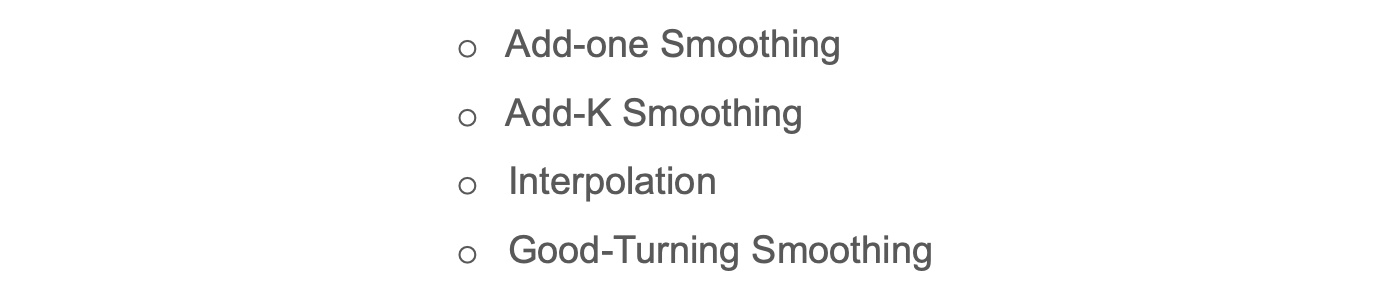
所以，perplexity是越小越好。



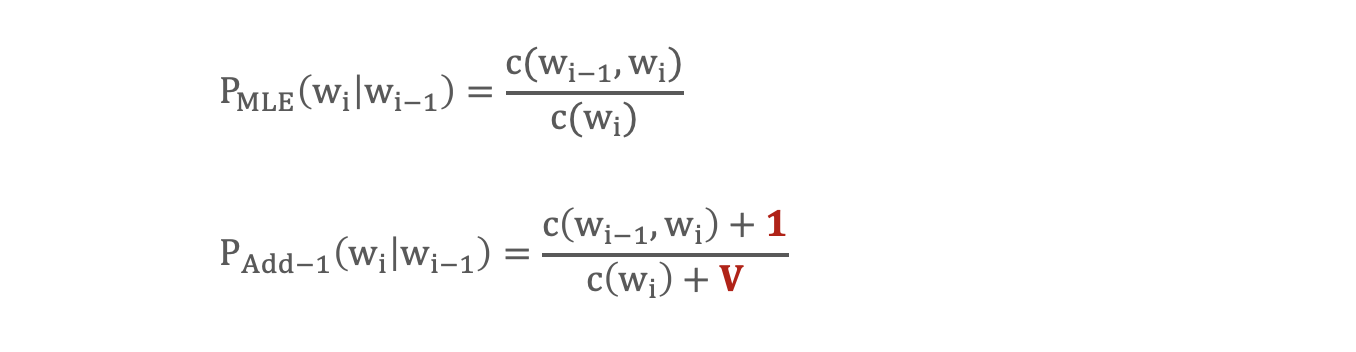


**5. Smoothing**

四类：



**5.1 Add-one Smoothing (Laplace Smoothing)**



MLE：只用“已看到的现象”去估计。

V：词典库的大小。

图片包含 文字

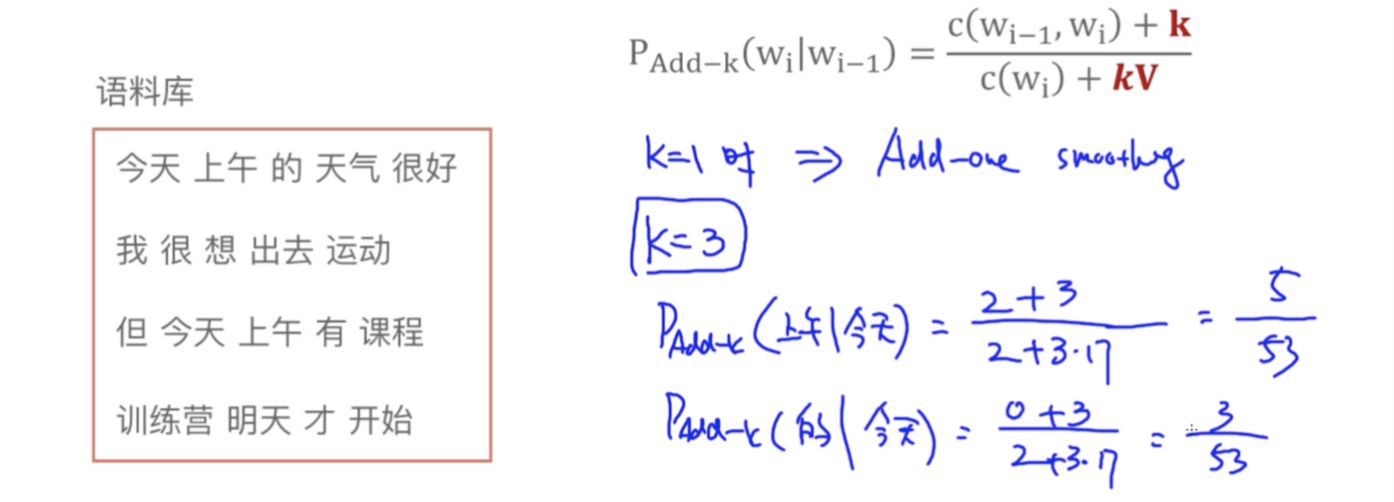
描述已自动生成

为什么分母+V：保证所有概率加起来=1。

**5.2 Add-K Smoothing (Laplace Smoothing)**

Add1是add-k的一种特例。

K相当于模型中的超参数，可以人工调整。（也可以让机器帮我们选择最优的）



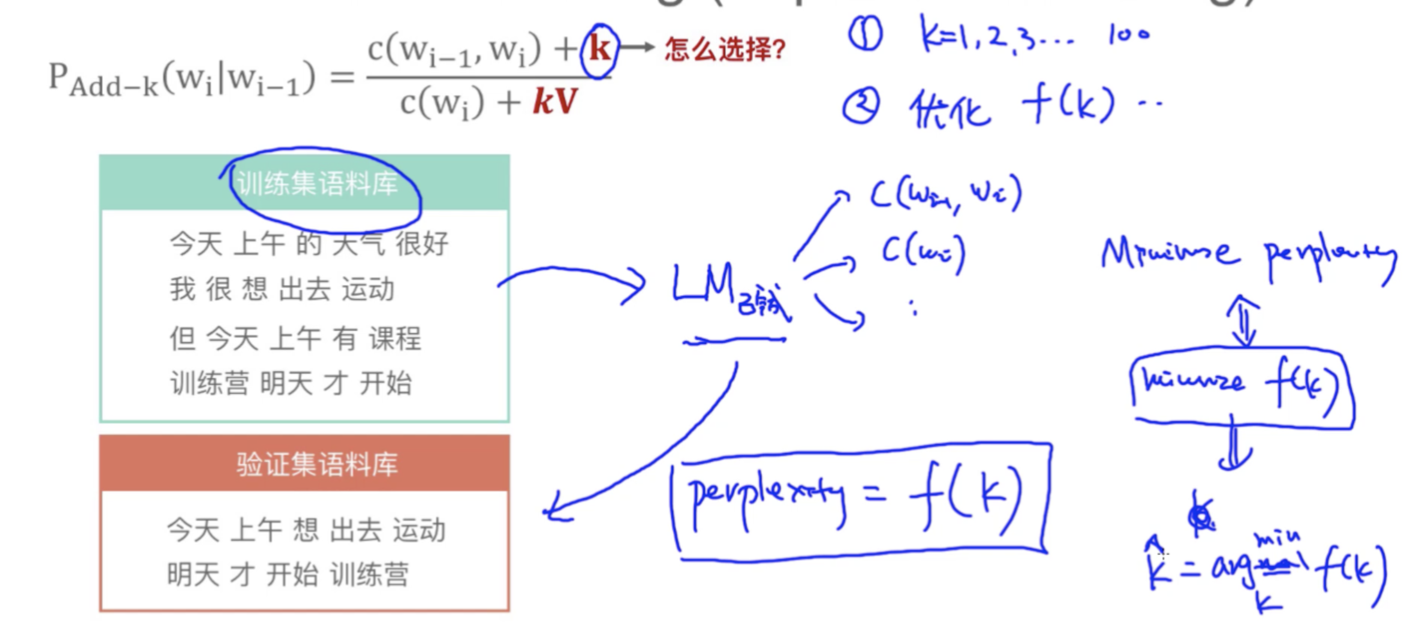
如何选择k？

1）k=1，2…手动选择调整。

2）优化f（k）

把已训练好的LM，去运用在验证集中，计算perplexity，它是关于k的一个函数。

因为好的LM的perplexity在验证集中越小越好，所以目标：min f（k）



**5.3** **Interpolation**

为什么提出Interpolation？

图片包含 文字

描述已自动生成

从已有的数据中，我们知道arboretum概率为0，kitchen出现过。但只是因为in the没有出现，所以两者概率都为0，这是不合理的。

因为，在未来的、未知的、扩大的数据集中，kitchen出现次数应该是更多的。

核心：在n-gram条件概率的模型下，把unigram/bigram也考虑进来。

