# 3-lm [>>](marginnote3app://note/A72B9904-1F4C-410D-AE0F-610C43FF377A)

## Language Modeling [>>](marginnote3app://note/790D2BA6-BF06-4A8D-ADFF-7110BFB6A434)

### Introduction to N-grams [>>](marginnote3app://note/C66CD79D-F5B5-401D-A24E-797D1E86CBEE)

#### 概率语言模型 [>>](marginnote3app://note/F98F89F1-4A2E-4B84-AD22-0745B7C99065) Probabilistic Language Models

##### [>>](marginnote3app://note/B01F1E4D-17D2-4A61-B0CA-8775D3937386)

- 目标：计算一个句子或词语序列的概率。P(W) = P(w1,w2,w3,w4,w5...wn)
- 相关任务：即将出现的单词的概率：P(w5|w1,w2,w3,w4)
- 一个能计算出其中任何一个的模型。P(W)或P(wn|w1,w2...wn-1)被称为一个语言模型。
- 更好：语法 但语言模型或LM是标准的

#### 如何计算P（W） [>>](marginnote3app://note/72EC0C68-C00E-4EAE-BEC7-713F868852AA) How to compute P(W)

##### Chain Rule of Probability [>>](marginnote3app://note/0637838D-5C0D-4712-B9E5-8495B558306C)

#### Reminder: The Chain Rule [>>](marginnote3app://note/71B10C74-83CE-4798-BF86-7B1E12782115)

##### [>>](marginnote3app://note/67E54DD9-AAB1-4811-BD90-9001F73E7FA1)

##### [>>](marginnote3app://note/AEA81839-4EBB-475D-87A5-99F108134A58)

#### 如何估计这些概率 [>>](marginnote3app://note/CF64AC4C-B9C7-4E0D-A95B-30FF864A99FE) How to estimate these probabilities

##### 我们能不能只计算和划分？- 不可以! 太多可能的句子了! - 我们永远不会看到足够的数据来估计这些 [>>](marginnote3app://note/6768F427-E39F-4F40-9B3E-63E951329D8E)

#### 马尔科夫假设 [>>](marginnote3app://note/3EFF871E-971F-474F-A47B-0C0FEB1086BA) Markov Assumption

##### [>>](marginnote3app://note/D251795D-8571-49A6-891D-E46130A0452D)

#### 最简单的案例 [>>](marginnote3app://note/44698072-BFBF-496C-9E6F-BD677FEDF893) Simplest case

##### Unigram model [>>](marginnote3app://note/5A374C7D-3835-4DF3-8DF5-14936EF2384C)

##### Bigram model [>>](marginnote3app://note/D1889C04-6A82-4953-BAC1-A2A42AF4234F)

##### N-gram models [>>](marginnote3app://note/04BC3B3B-0B9D-46FA-B852-B9D7CA9CA17F)

### 估计N-gram概率 [>>](marginnote3app://note/AC57B4BF-C94E-4894-B80F-FDD024910342) Estimating N-gram Probabilities

#### Estimating bigram probabilities [>>](marginnote3app://note/95EEBCC0-5FCF-4646-B181-8A3D7F0EC783)

##### [>>](marginnote3app://note/7589691A-46FC-473A-BE34-F7B04E953ED4)

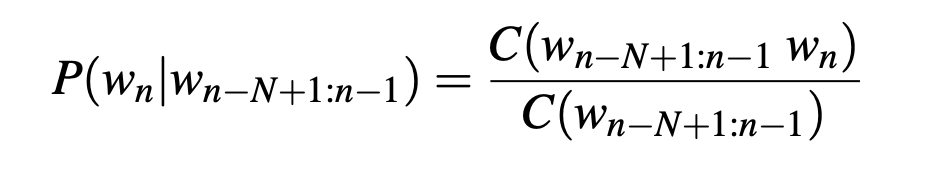
#### An example [>>](marginnote3app://note/A99891CB-F80E-4A78-8670-9BC27608C685)

##### [>>](marginnote3app://note/AC989559-BE2F-433B-BD95-CF9527EC6E89)

首先需要在每个句子的开头加上一个特殊的符号<s>，以便为我们提供第一个词的bigram线索。我们还需要一个特殊的结束符号。</s>  
1 对于概率模型来说，归一化意味着除以某个总计数，使所得概率合法地落在0和1之间。
2 我们需要末尾符号来使大词语法成为真正的概率分布。如果没有结束符，所有给定长度的句子的概率之和将为1。这个模型将定义一个无限的概率分布集，每个句子长度有一个分布。

#### More examples: Berkeley Restaurant Project sentences [>>](marginnote3app://note/E0D69502-1644-4890-8101-EC0136CDFC02)

##### [>>](marginnote3app://note/B38F3FEB-D302-4E3A-9823-5CF58F0CDD35)

  
公式跟上面公式一样通过将特定序列的观察频率除以前缀的观察频率来估计n-gram概率。
这个比率被称为相对频率。我们在上面说过，这种使用相对相对频率作为估计概率的方式是最大似然估计或MLE。

###### [>>](marginnote3app://note/08ADF6E1-17A6-4F06-98A8-04E8226E483D)

###### [>>](marginnote3app://note/B63B9A39-256D-4F92-9BF1-610ED3F448B6)

###### [>>](marginnote3app://note/67014FBE-C16A-4A72-A00B-3CF79C951BAF)

###### [>>](marginnote3app://note/E377F62B-CF29-4E18-841B-9E75560FFA90)

##### [>>](marginnote3app://note/215C593B-1F2C-471F-B264-F85F77664CB7)

### Evaluation and Perplexity [>>](marginnote3app://note/CF970D92-FA77-4A47-A193-62F2009F0C8E)

#### Evaluation [>>](marginnote3app://note/17C8105D-1CB9-4378-968A-4729A35D6C94)

##### Evaluation: How good is our model? [>>](marginnote3app://note/BF60C817-003A-4CB7-AEA2-708C3C932362)

我们的语言模型是否喜欢好的句子而不是坏的句子？
- 对 "真实的 "或 "经常观察到的 "句子赋予更高的概率
-而不是 "不合语法的 "或 "很少观察到的 "句子？
- 我们在训练集上训练我们模型的参数。
- 我们在未见过的数据上测试该模型的性能。
- 测试集是一个未见过的数据集，与我们的训练集不同，完全没有使用。
- 一个评估指标告诉我们，我们的模型在测试集上的表现如何

##### 在测试集上进行训练 [>>](marginnote3app://note/C8413742-D048-4A09-B537-985C2BC48733) Training on the test set

-我们不能让测试句子进入训练集
-当我们把它设置在测试集时，我们会人为地赋予它一个高的概率
-"在测试集上训练"
-糟糕的科学!
- 而且违反了荣誉守则

##### N-gram模型的外在评价 [>>](marginnote3app://note/968580C6-F350-46A0-96C0-2065D9858D6C) Extrinsic evaluation of N-gram models

比较模型A和B的最佳评估
- 把每个模型放在一个任务中
- 拼写纠正器、语音识别器、MT系统
- 运行任务，得到A和B的准确度
- 有多少拼写错误的词被正确纠正 - 有多少词被正确翻译
- 比较A和B的准确度

##### N-gram模型的外在（体内）评价的困难性 [>>](marginnote3app://note/E5C1FF25-C4AB-4B06-93E3-8DDF3AC17558) Difficulty of extrinsic (in-vivo) evaluation of N-gram models

外在评价--耗时；可能需要几天或几周--所以--有时使用内在评价：迷惑性--不好的近似--除非测试数据看起来和训练数据一样--所以一般只在试点实验中有用--但对思考有帮助。

#### Perplexity [>>](marginnote3app://note/49B0F295-FA6E-461A-AF07-27D479DBC004)

在实践中,我们不使用原始概率作为评估语言模型的度量标准,而是使用一种称为perplexity的变体。测试集上语言模型的perplexity(有时简称PP)是测试集的逆概率,由单词的数量标准化。

##### [>>](marginnote3app://note/C3695B56-59B5-46A1-A46B-DF773072C272)

##### [>>](marginnote3app://note/0F7F4347-FAC3-469B-AA35-EAFED56F8344)

##### Lower perplexity = better model [>>](marginnote3app://note/BBCA5C38-E1E2-40A7-9877-A164966AC927)

### 泛化和零点 [>>](marginnote3app://note/187800FC-2729-4996-8A95-1F1131E6E1AD) Generalization and zeros

#### The Shannon Visualization Method [>>](marginnote3app://note/A4802E92-1A8E-4D95-A007-3EA9826382F7)

N-grams只有在测试语料库与训练语料库相似的情况下才能很好地用于单词预测 - 在现实生活中，往往不是这样的 - 我们需要训练能够泛化的稳健模型!

#### 一种泛化。零点 - 在训练集中不曾出现的东西 - 但在测试集中会出现 [>>](marginnote3app://note/5DFFC71C-4EDB-4F90-9FC5-B5811DCC68DE)

##### 概率为零的biggrams—意味着我们将给测试集分配0的概率！因此我们不能计算perplexisty（不能除以0）。 [>>](marginnote3app://note/8A347132-B42F-4026-B27B-78A70C51BD8F)

### 平滑化：加一（拉普拉斯）平滑化 [>>](marginnote3app://note/73E1C64F-773C-4483-AABF-9EBE85061277) Smoothing: Add-one (Laplace) smoothing

#### 加一估计 [>>](marginnote3app://note/F7430853-AC70-4088-A76F-334C87AFE0C7) Add-one estimation

Add-1的估计是一个钝器 - 所以add-1不用于N-grams。- 我们会看到更好的方法 - 但是add-1被用来平滑其他NLP模型 - 用于文本分类 - 在零的数量不那么大的领域。

##### [>>](marginnote3app://note/8C575572-38D6-4776-85A3-76C33BA7D6A3)

#### 最大似然估计 [>>](marginnote3app://note/101B119E-CF76-486D-9FB5-8F8C63F84BEF) Maximum Likelihood Estimates

##### 最大似然估计--来自训练集T的模型M的某些参数--使训练集T在模型M下的似然最大化 -假设 "bagel "这个词在一百万字的语料库中出现了400次 -其他文本中的随机词是 "bagel "的概率是多少？-MLE的估计值是4001,000,000=0.0004 -对于其他一些语料库来说，这可能是一个糟糕的估计值 -但这是一个使 "贝果 "在一百万字的语料库中出现400次的最有可能的估计值。 [>>](marginnote3app://note/5DC2F743-623E-4B9F-BA9B-3742DA07A97F)

#### Berkeley Restaurant Corpus: Laplace smoothed bigram counts [>>](marginnote3app://note/087913F2-CAFE-4D0C-8D20-BE0184F24F59)

##### [>>](marginnote3app://note/9CB68980-1BD6-4024-983E-22D07A4D145E)

##### [>>](marginnote3app://note/D7D3BD6C-E2F1-43C1-B553-F8C6648F47CA)

##### [>>](marginnote3app://note/666C1763-0B91-4890-8E9B-1BD6140541F3)

### 插值、倒退和网络规模的LMs [>>](marginnote3app://note/6F8B0D59-7B5C-4A71-AE26-085875F68851) Interpolation, Backoff, and Web-Scale LMs

#### 倒退和插值 [>>](marginnote3app://note/7A5A3A0A-72C2-494C-9E41-992A1157C127) Backoff and Interpolation

有时，使用较少的语境是有帮助的--对于你还没有学到很多东西的语境，条件是较少的语境。

##### Backoff: • use trigram if you have good evidence, • otherwise bigram, otherwise unigram [>>](marginnote3app://note/8D104C1C-6873-461B-8743-5FA5AD11D40A)

倒退。- 如果你有很好的证据，就用trigram，否则bigram，否则用unigram

##### Interpolation: • mix unigram, bigram, trigram [>>](marginnote3app://note/111D771A-96B4-440E-BF93-748491031613)

内插法效果更好

#### 线性插值 [>>](marginnote3app://note/15C2CE3F-5425-401B-ADC0-425C01DEF226) Linear Interpolation

##### [>>](marginnote3app://note/4B15A8CF-80AD-4FBF-A3D8-B5B543F3A04F)

#### How to set the lambdas? [>>](marginnote3app://note/49BD9E9D-0963-444F-93D6-BC0FA095168D)

##### [>>](marginnote3app://note/7B13755D-E883-4927-AB72-3A3168A5C19E)

选择λs以使保留的数据的概率最大化。- 固定N-gram的概率（在训练数据上） - 然后寻找能使保留数据集的概率最大的λs。

#### Advanced Language Modeling [>>](marginnote3app://note/329460BD-B1FA-49B3-9220-34F74989AC29)

##### [>>](marginnote3app://note/95F15582-6F56-428B-B983-FA5594CA0FAF)

#### Smoothing for Web-scale N-grams [>>](marginnote3app://note/4A837211-E9D1-4E5C-830B-E23AA4380358)

网络规模N-grams的平滑化

##### [>>](marginnote3app://note/D1A8B063-0C80-43F7-9F29-0D21ABD6B343)

#### 巨大的网络规模n-grams [>>](marginnote3app://note/C2F9E040-7F35-4F42-AF8A-37AFF7FE76E8) Huge web-scale n-grams

如何处理，例如，谷歌N-gram语料库
修剪
- 只存储计数大于阈值的N-grams。
- 删除高阶N-grams的单子
- 基于熵的修剪
效率
- 高效的数据结构，如tries
- 布隆过滤器：近似的语言模型
- 将单词存储为索引，而不是字符串
- 使用Huffman编码将大量单词装入两个字节
- 量化概率（4-8位，而不是8字节的浮点）

#### 未知词。开放式与封闭式的词汇任务 [>>](marginnote3app://note/72E8C8B0-9AAF-42CD-8230-7BC909B3460B) Unknown words: Open versus closed vocabulary tasks

如果我们知道所有的高级词汇
- 词汇表V是固定的
- 封闭式词汇任务
通常我们不知道
- 词汇表外=OOV词汇
- 开放式词汇任务
相反：创建一个未知的单词标记<UNK>
- 训练<UNK>概率
- 创建一个大小为V的固定词库L
- 在文本规范化阶段，任何不在L中的训练词都被改为<UNK>
- 现在我们像正常单词一样训练其概率
在解码时间
- 如果文本输入。对任何不在训练中的词使用UNK概率

# ed3book\_dec302020 [>>](marginnote3app://note/C0815E4B-5A1E-4BD0-B82F-E78C5BD3436C)

# N-gram 语言模型 - 知乎\_87207dc6f9cd3be7f6acb30847ec37e9 [>>](marginnote3app://note/AC6440A1-E9F1-4BEC-B8F0-E38273F66CF7)

# N-gram 语言模型 - 知乎\_87207dc6f9cd3be7f6acb30847ec37e9 [>>](marginnote3app://note/66703B93-19C5-4D10-A5D9-E7FBD0C91D69)