

1. Noisy Channel Model

$$p(\text{text}|\text{source}) \propto p(\text{source}|\text{text}) p(\text{text})$$

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。

机器翻译 i.e. 英 → 中

$$P(\text{中文}|\text{英文}) \propto P(\text{英文}|\text{中文}) \cdot P(\text{中文}) \rightarrow \text{语言模型}$$

argmax ↓ Translation

拼写纠错

$$P(\text{正确的写法}|\text{错误的写法}) \propto P(\text{错误的写法}|\text{正确的写法}) \cdot P(\text{正确的写法}) \rightarrow \text{语言模型}$$

语音识别 输入: ~~~~~

$$P(\text{文本}|\text{语音信号}) \propto P(\text{语音信号}|\text{文本}) \cdot P(\text{文本}) \rightarrow \text{语言模型}$$

Translation
Recognition model

密码破解 输入: encrypted string (abffde...)

$$P(\text{明文}|\text{暗文}) \propto P(\text{暗文}|\text{明文}) \cdot P(\text{明文}) \rightarrow \text{语言模型}$$

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

目标

Compute the probability of a sentence or sequence of words. $p(s) = p(w_1, w_2, w_3, w_4, w_5 \dots w_n)$

2.1 Chain Rule

Random Variable Recap: Chain Rule

$$\begin{aligned} P(A, B) &= P(A|B) \cdot P(B) \\ &= P(B|A) \cdot P(A) \end{aligned}$$

$$\circ p(A, B, C, D) = P(A) \cdot P(B|A) \cdot P(C|A, B) \cdot P(D|A, B, C) \leftarrow \text{chain rule}^1$$

$$= P(A, B) \cdot P(C|A, B) \cdot P(D|A, B, C)$$

$$= P(A, B, C) \cdot P(D|A, B, C) = P(A, B, C, D)$$

$$\circ p(w_1, w_2, w_3, w_4, w_5 \dots w_n)$$

$$= P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1, w_2) \cdot P(w_4|w_1, w_2, w_3) \cdots P(w_n|w_1, w_2, \dots, w_{n-1})$$

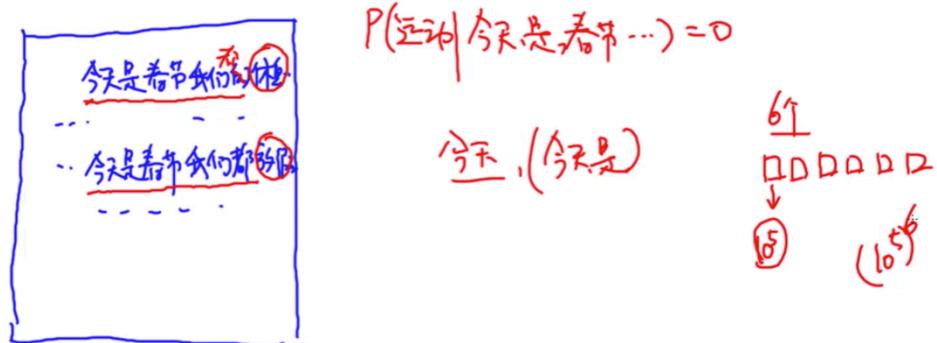
$$\cdots \cdots P(w_n|w_1, w_2, \dots, w_{n-1}) \leftarrow \text{chain rule}$$

$$\circ p(\text{今天}, \text{是}, \text{春节}, \text{我们}, \text{都}, \text{休息})$$

$$\begin{aligned} &= P(\text{今天}) \cdot P(\text{是}|\text{今天}) \cdot P(\text{春节}|\text{今天, 是}) \cdot P(\text{我们}|\text{今天, 是, 春节}) \\ &\quad \cdot P(\text{都}|\text{今天, 是, 春节, 我们}) \cdot P(\text{休息}|\text{今天, 是, 春节, 我们, 都}) \end{aligned}$$

2.2 Chain Rule for Language Model

$$\circ p(\text{休息} | \text{今天, 是, 春节, 我们, 都}) \approx 0 \quad \underline{\text{Sparsity}}$$



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

$$\circ p(\text{休息} | \text{今天, 是, 春节, 我们, 都}) \approx p(\text{休息} | \text{都}) \rightarrow \text{1st order markov assumption}$$

$$\circ p(\text{休息} | \text{今天, 是, 春节, 我们, 都}) \approx p(\text{休息} | \text{我们, 都}) \downarrow \text{2nd order markov assumption}$$

$$\circ p(\text{休息} | \text{今天, 是, 春节, 我们, 都}) \approx p(\text{休息} | \text{春节, 我们, 都}) \downarrow \text{3rd order ...}$$

- $p(w_1, w_2, w_3, w_4, w_5 \dots w_n)$
- $= p(w_1) \cdot p(w_2|w_1) \cdot p(w_3|w_2) \dots p(w_n|w_{n-1}) = p(w_1) \prod_{i=2}^n p(w_i|w_{i-1})$ 1st order
- $p(w_1, w_2, w_3, w_4, w_5 \dots w_n)$
- $= p(w_1) \cdot p(w_2|w_1) \cdot p(w_3|w_1, w_2) \cdot p(w_4|w_2, w_3) \dots p(w_n|w_{n-2}, w_{n-1})$ 2nd order
- $p(w_1, w_2, w_3, w_4, w_5 \dots w_n)$
- $= p(w_1) \cdot p(w_2|w_1) \cdot \prod_{i=3}^n p(w_i|w_{i-2}, w_{i-1})$ 3rd order

Language Model (Use 2nd Order)

LM

$$\begin{aligned} p(\text{是}|\text{今天}) &= 0.01 \\ p(\text{今天}) &= 0.002 \\ p(\text{周日}|\text{是}) &= 0.001 \\ p(\text{周日}|\text{今天}) &= 0.0001 \\ p(\text{周日}) &= 0.02, \\ p(\text{是}|\text{周日}) &= 0.0002 \end{aligned}$$

比较: 今天是周日 VS 今天周日是

$$\begin{aligned} P_{\text{LM}}(\text{今天是周日}) &= P_{\text{LM}}(\text{今天}) \cdot P(\text{周日}|\text{是}) \\ &= 0.002 \cdot 0.01 \cdot 0.001 \\ &= 2 \times 10^{-8} \end{aligned} \quad > \quad \begin{aligned} P_{\text{LM}}(\text{今天周日是}) &= P_{\text{LM}}(\text{今天}) \cdot P(\text{周日}|\text{今天}) \\ &\cdot P(\text{是}|\text{周日}) \\ &= 0.002 \cdot 0.0001 \cdot 0.002 \\ &= 4 \times 10^{-10} \end{aligned}$$

2.4 Classification

1) unigram

- $p(w_1, w_2, w_3, w_4, w_5 \dots w_n)$
- $= p(w_1) \cdot p(w_2|w_1) \cdot p(w_3|w_2) \dots p(w_n|w_{n-1})$
- $p(\text{今天, 是, 春节, 我们, 都, 休息})$ ✓
- $= p(\text{今天}) \cdot p(\text{是}) \cdot p(\text{春节}) \cdot p(\text{我们}) \cdot p(\text{都}) \cdot p(\text{休息})$
- $p(\text{今天, 春节, 是, 都, 我们, 休息})$ -
- $= p(\text{今天}) \cdot p(\text{春节}) \cdot p(\text{是}) \cdot p(\text{都}) \cdot p(\text{我们}) \cdot p(\text{休息})$

2) bigram

Language Model : Bigram

\leftarrow 1st order
markov assumption

- $p(w_1, w_2, w_3, w_4, w_5 \dots w_n)$
 $= p(w_1) \cdot p(w_2|w_1) \cdot p(w_3|w_2) \dots p(w_n|w_{n-1}) = p(w_1) \prod_{i=2}^n p(w_i|w_{i-1})$
- $p(\text{今天}, \text{是}, \text{春节}, \text{我们}, \text{都}, \text{休息})$
 $= p(\text{今天}) \cdot p(\text{是}|\text{今天}) \cdot p(\text{春节}|\text{是}) \cdot p(\text{我们}|\text{春节}) \cdot p(\text{都}|\text{我们}) \cdot p(\text{休息}|\text{都})$
- $p(\text{今天}, \text{春节}, \text{是}, \text{都}, \text{我们}, \text{休息})$
 $= p(\text{今天}) \cdot p(\text{春节}|\text{今天}) \cdot p(\text{是}|\text{春节}) \cdot p(\text{都}|\text{是}) \cdot p(\text{我们}|\text{都}) \cdot p(\text{休息}|\text{我们})$

3) ...N-gram

Language Model : N-gram

\leftarrow > 2
Higher Order

$N=3$

- $p(w_1, w_2, w_3, w_4, w_5 \dots w_n)$
 $= p(w_1) \cdot p(w_2|w_1) \cdot p(w_3|w_1, w_2) \cdot p(w_4|w_1, w_2, w_3) \dots p(w_n|w_1, w_2, \dots, w_{n-1}) = p(w_1) \prod_{i=2}^n p(w_i|w_{i-1}, \dots, w_1)$
- $p(\text{今天}, \text{是}, \text{春节}, \text{我们}, \text{都}, \text{休息})$
 $= p(\text{今天}) \cdot p(\text{是}|\text{今天}) \cdot p(\text{春节}|\text{今天是}) \cdot p(\text{我们}|\text{是春节}) \dots$

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

$$\begin{aligned}
 & \circ p(w_1, w_2, w_3, w_4, w_5 \dots w_n) \\
 & = p(w_1) \cdot p(w_2|w_1) \cdot p(w_3|w_1, w_2) \dots p(w_n|w_1, w_2, \dots, w_{n-1}) \\
 & p(\text{是}|\text{明天}) \quad \boxed{\begin{array}{c} \text{明天是} \\ \text{明天我们} \\ \text{明天上课} \\ \text{明天是} \\ \text{明天天气} \end{array}} \quad p(\text{是}|\text{明天}) = \frac{2}{5} \\
 & p(\text{我们}|\text{明天}) = \frac{1}{5} \\
 & p(\text{上课}|\text{明天}) = p(\text{天气}|\text{明天}) = \frac{1}{5}
 \end{aligned}$$

语料库

今天的天气很好啊
我很想出去运动
但今天上午想上课
训练营明天才开始

V=19

$$\begin{aligned} P_{\text{lm}}(\text{今天 上午 想 出去 运动}) \\ = \underbrace{P_{\text{lm}}(\text{今})}_{\frac{2}{19}} \cdot \underbrace{P_{\text{lm}}(\text{上}|\text{今})}_{\frac{1}{2}} \cdot \underbrace{P_{\text{lm}}(\text{想}|\text{上}今})}_{1} \cdot \underbrace{P_{\text{lm}}(\text{出}|\text{想上今})}_{1} \\ = \frac{2}{19} \cdot \frac{1}{2} \cdot 1 \cdot \frac{1}{2} \cdot 1 = \frac{1}{38} \quad P_{\text{lm}}(\text{好}|\text{出}) \\ P_{\text{lm}}(\text{今天 上午 的 天气 很好 呢}) \\ = P_{\text{lm}}(\text{今}) \cdot P_{\text{lm}}(\text{上}|\text{今}) \cdot P_{\text{lm}}(\text{好}|\text{上今}) \cdot \dots \\ = 0 \end{aligned}$$

4. Evaluation of Language Model

Q: 训练出来的语言模型效果好还是坏?

- 理想情况下
 1. 假设有两个语言模型 A,B
 2. 选定一个特定的任务比如拼写纠错
 3. 把两个模型A,B都应用在此任务中
 4. 最后比较准确率，从而判断A,B的表现

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)

核心思路

今天____
今天天气____,
今天天气很好, ____
今天天气很好, 适合____
今天天气很好, 适合出去____

4.1 Perplexity

尤其是在无监督学习下，通常使用 perplexity 来评估。

$$\text{Perplexity} = 2^{-(x)} \quad x: \text{average log likelihood}$$

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。

训练好的 Bigram

$$\begin{aligned} p(\text{天气}|\text{今天}) &= 0.01 \\ p(\text{今天}) &= 0.002 \\ p(\text{很好}|\text{天气}) &= 0.1 \\ p(\text{适合}|\text{很好}) &= 0.01 \\ p(\text{出去}|\text{适合}) &= 0.02, \\ p(\text{运动}|\text{出去}) &= 0.1 \end{aligned}$$

$$\text{今天 } P(\text{今天}) = 0.002 \Rightarrow \log P(\text{今天}) = \alpha_1$$

$$\text{今天 天气 } P(\text{天气}|\text{今天}) = 0.01 \Rightarrow \log P(\text{天气}|\text{今天}) = -2$$

$$\text{今天 天气 很好, } P(\text{很好}|\text{天气}) = 0.1 \Rightarrow \log P(\text{很好}|\text{天气}) = -1$$

$$\text{今天 天气 很好, 适合 } P(\text{适合}|\text{很好}) = 0.01 \Rightarrow -2$$

$$\text{今天 天气 很好, 适合 出去 } P(\text{出去}|\text{适合}) = 0.02 = \alpha_2$$

$$\text{今天 天气 很好, 适合 出去 运动 } P(\text{运动}|\text{出去}) = \alpha_1 = -1$$

Training 38 million words, test 1.5 million words, WSJ

N-gram Order	Unigram	Bigram	Trigram	
Perplexity	962	170	109	}

5. Smoothing

四类：

- Add-one Smoothing
- Add-K Smoothing
- Interpolation
- Good-Turning Smoothing

5.1 Add-one Smoothing (Laplace Smoothing)

$$P_{MLE}(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_i)}$$

$$P_{Add-1}(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i) + 1}{c(w_i) + V}$$

MLE：只用“已看到的现象”去估计。

V：词典库的大小。

$$V = 17$$

语料库

今天 上午 的 天气 很好
我 很 想 出去 运动
但 今 天 上午 有 课 程
训 练 营 明 天 才 开 始

$$P_{Add-1}(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i) + 1}{c(w_i) + V}$$

$$\begin{aligned} P_{Add-1}(\text{上午}|\text{今天}) &= \frac{2+1}{2+17} = \frac{3}{19} \\ P_{Add-1}(\text{很好}|\text{今天}) &= \frac{0+1}{2+17} = \frac{1}{19} \\ P_{Add-1}(\text{运动}|\text{今天}) &= \frac{1}{19} \\ &\vdots \\ P_{Add-1}(\text{开始}|\text{今天}) &= \frac{1}{19} \end{aligned} \quad \left. \begin{array}{l} \frac{3}{19} + \frac{1}{19} \\ = \frac{4}{19} \\ = 1 \end{array} \right\}$$

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）

$$P_{\text{Add}-k}(w_i | w_{i-1}) = \frac{c(w_{i-1}, w_i) + k}{c(w_i) + kV}$$

语料库

今天 上午 的 天气 很好
我 很 想 出去 运动
但 今 天 上午 有 课 程
训练营 明天 才 开始

$k=1$ 时 \Rightarrow Add-one smoothing

$\boxed{k=3}$

$$P_{\text{Add}-k}(\text{上午} | \text{今天}) = \frac{2+3}{2+3 \cdot 1} = \frac{5}{53}$$

$$P_{\text{Add}-k}(\text{好} | \text{今天}) = \frac{0+3}{2+3 \cdot 1} = \frac{3}{53}$$

如何选择 k？

1) $k=1, 2, \dots$ 手动选择调整。

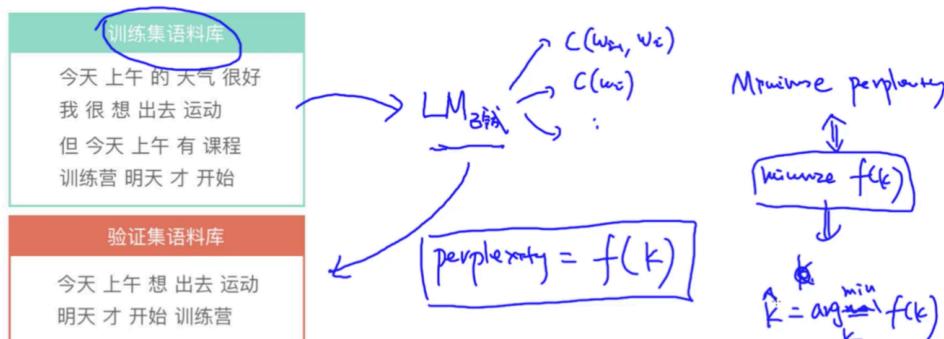
2) 优化 $f(k)$

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： $\min f(k)$

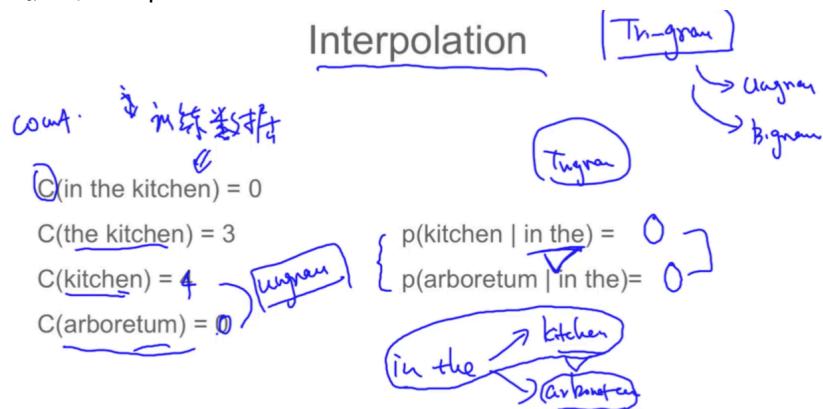
$$P_{\text{Add}-k}(w_i | w_{i-1}) = \frac{c(w_{i-1}, w_i) + k}{c(w_i) + kV} \rightarrow \text{怎么选择?}$$

① $k=1, 2, 3, \dots, 100$
② 优化 $f(k)$..



5.3 Interpolation

为什么提出 Interpolation？



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。

核心思路

在计算Trigram概率时同时考虑Unigram,
Bigram, Trigram出现的频次

$$\begin{aligned} p(w_n | w_{n-1}, w_{n-2}) &= \lambda_1 p(w_n | w_{n-1}, w_{n-2}) \\ &\quad + \lambda_2 p(w_n | w_{n-1}) \\ &\quad + \lambda_3 p(w_n) \end{aligned}$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

5.4 Good-Turning Smoothing

假设你在钓鱼，已经抓到了18只鱼：

10条鲤鱼，3条黑鱼，2条刀鱼，1条鲨鱼，1条草鱼，1条鳗鱼.....

Q1: 下一个钓到的鱼是鲨鱼的概率是多少？

Q2: 下一条鱼是新鱼种（之前没有出现过）的概率是多少？

Q3: 既然如此，重新想一下，下一条抓到的鱼为鲨鱼的概率是多少？

Q1: 下一个钓到的鱼是鲨鱼的概率是多少？ $\frac{1}{18}$  = 1

Q2: 下一条鱼是新鱼种（之前没有出现过）的概率是多少？ $\frac{3}{18}$

Q3: 既然如此，重新想一下，下一条抓到的鱼为鲨鱼的概率是多少？ $\frac{1}{18}$  新鱼 < 1

Q1: 我们是假设六种鱼占据了所有的概率空间。

Q2: 我们需要把一些空间留给新鱼种。

Q3: 那么，在考虑了预留空间后，鲨鱼的概率必定就变小了。

定义一个变量：**Nc**，出现 c 次的单词的个数。

N_c 出现c次的单词的个数

例: Sam I am I am Sam I do not eat

Sam :	2次 ✓	$N_3 = 1$: 有多少单词出现了3次?
I :	3次	$N_2 = 3$
am :	2次 ✓	$N_1 = 3$
Sam :	2次 ✓	
do :	1次	
not :	1次	
eat :	1次	就是这样 的意思

没有出现过的单词

$$P_{MLE} = 0$$

$$P_{GT} = \frac{N_1}{N}$$

假设你在钓鱼, 已经抓到了18只鱼:

10条鲤鱼, 3条黑鱼, 2条刀鱼, 1条鲨鱼, 1条草鱼, 1条鳗鱼.....

出现过的单词

$$P_{MLE} = \frac{c}{N}$$

$$P_{GT} = \frac{(c+1)N_{c+1}}{N_c}$$

MLE: 我看到什么, 就预测什么, 不考虑未来。

没有出现过的

$$P_{MLE}(\text{黑鱼}) = \frac{0}{18} = 0$$

$$P_{GT}(\text{黑鱼}) = \frac{1}{N} = \frac{3}{18} = \frac{1}{6}$$

出现过的

$$P_{MLE}(\text{草鱼}) = \frac{1}{18}$$

$$= \frac{2 \cdot 1}{3 \cdot 18}$$

$$P_{GT}(\text{草鱼}) = \frac{(1+1)N_2}{N_c N} = \frac{2 \cdot 1}{3 \cdot 18}$$

Count r	Count of counts N_r	Adjusted count r^*	Test count t
0	7,514,941,065.	0.00015	0.00016
1	1,132,844	0.46539	0.46235
2	263,611	1.40679	1.39946
3	123,615	2.38767	2.34307
4	73,788	3.33753	3.35202
5	49,254	4.36967	4.35234
6	35,869	5.32928	5.33762
8	21,693	7.43798	7.15074
10	14,880	9.31304	9.11927
20	4,546	19.54487	18.95948

缺点：

例如，我们需要计算期望出现 101 次的单词的数据，我们需要出现 102 次单词的数据。
但我们不能确保一定有单词出现 102 次

解决方法：用图像、机器去拟合这个趋势。

6. Generate Sentence

LM 是生成模型，也就是可以生成一些新的数据，例如小说、图片等。

(还有判别模型)

6.1 unigram

[NLP, I, like, studying, course, yesterday] = Vocab
 0.1 0.3 0.2 0.2 0.35 0.05 ←

概率
统计
分布

(I, study, NLP, course, I, yesterday)

Unigram 是不考虑上下文信息的，只是相当于用这些概率来随机选择单词，组成一句话。
一般是不符合语言逻辑的

6.2 bigram

[NLP, I, like, studying, course, yesterday]



会有一个概率矩阵。

[NLP, I, like, studying, course, yesterday]

MLP I (like study course yesterday)

MP
I
like
study
course
yesterday

①

	0.01	0.01	0.01	0.01	0.99	0.01	-
I	0.01	0.01	0.95	0.01	0.01	0.99	-
like	0.4	0	0.5	0.1	0	-	-
study	0.3	-	-	-	0.4	0.2	-
course	-	-	-	-	-	0.95	-
yesterday	-	-	-	-	-	-	0.99

I like studying course yesterday ↓
Terminate +

所以这是一个很简单的一个思路