n元模型

常宝宝
北京大学计算语言学研究所
chbb@pku.edu.cn

语言建模(Language Modeling)

- 从统计角度看，自然语言中的句子s可由任何词串构成，但概率不同。如：

$$s_1 = \text{我 刚 吃 过 晚饭}$$
$$s_2 = \text{刚 我 过 晚饭 吃}$$
$$P(s_1) > P(s_2)$$

- 给定自然语言L，$P(s)$未知
- 利用给定的语言样本估计$P(s)$的过程被称作语言建模。

语言模型(language model)

- 根据语言样本估计出的概率分布P称为语言L的语言模型。

$$\sum_{s \in L} P(s) = 1$$

- 语言模型给句子赋以概率
  - s作为语言L中句子的概率
- 把数学方法引入符号系统

语言模型

- 语音识别

  I have **too many** books. (√)

  I have **to many** books.   (×)

  I have **two many** books. (×)

- 汉语分词

  别 **把** **手** 伸 进 别人 的 口袋 里(√)

  别 **把手** 伸 进 别人 的 口袋 里(×)

- 机器翻译

  我喜欢吃苹果 ⇒

  I like eating apple (√)

  I eating like apple (×)

语言建模

- 给定句子$s = w_1 w_2 \dots w_l$，如何计算$P(s)$?
  – 统计语料库中句子s出现的次数

- 应用链式规则，分解计算$P(s)$

$$P(s) = P(w_1)P(w_2|w_1)P(w_3|w_1 w_2) \dots P(w_l|w_1 w_2 \dots w_{l-1})$$
$$= \prod_{i=1}^{l} P(w_i|w_1 w_2 \dots w_{i-1})$$

$$P(John\ read\ a\ book)$$
$$= P(John) \times P(read|John) \times P(a|John\ read)$$
$$\times P(book|John\ read\ a)$$

n元模型

- **马尔可夫假设**(Markov assumption)
  w_i的出现只与之前的$n-1$个词有关
  $$P(w_i|w_1 w_2 \dots w_{i-1}) = P(w_i|w_{i-n+1} w_{i-n+2} \dots w_{i-1})$$

- 只需考虑n个词组成的片段，即n元组(n-gram)
  $$w_{i-n+1} w_{i-n+2} \dots w_{i-1} w_i$$

$$P(s) = P(w_1)P(w_2|w_1)P(w_3|w_1 w_2) \dots P(w_l|w_{l-n+1} w_{l-n+2} \dots w_{l-1})$$
$$= \prod_{i=1}^{l} P(w_i|w_{i-n+1} w_{i-n+2} \dots w_{i-1})$$

n元模型

n元模型

- 一元模型(n=1, unigram)

$$P(s) = P(w_1)P(w_2) \ldots P(w_i) \ldots P(w_l)$$

- 二元模型(n=2, bigram)

$$P(s) = P(w_1)P(w_2|w_1) \ldots P(w_i|w_{i-1}) \ldots . P(w_l|w_{l-1})$$

- 三元模型(n=3, trigram)

$$P(s) = P(w_1)P(w_2|w_1) \ldots P(w_i|w_{i-2}w_{i-1}) \ldots P(w_l|w_{l-2}w_{l-1})$$

$$P(John\ read\ a\ book)$$
$$= P(John) \times P(read|John) \times P(a|read) \times P(book|a)$$

n元模型的参数

	参数形式	参数数量
一元模型	$P(w_i)$	$\|V\|$
二元模型	$P(w_i\|w_{i-1})$	$\|V\| \times \|V\| = \|V\|^2$
三元模型	$P(w_i\|w_{i-2}w_{i-1})$	$\|V\| \times \|V\| \times \|V\| = \|V\|^3$
…	…	…
n元模型	$P(w_i\|w_{i-n+1} \dots w_{i-1})$	$\|V\| \times \cdots \times \|V\| = \|V\|^n$

注: $w \in V, V$ 代表词表，$|V|$代表词表中词的数量

n越大，模型需要的参数越多

参数数量指数增长

历史信息的作用

- 句子中前面出现的词对后面可能出现的词有很强的预示作用
- 词的历史信息对其后出现的词有选择限制作用：

a _______.   book or vegetable

John read a _______.   book or ~~vegetable~~

- 历史信息越多，选择限制越强。

the large green _______. pill, mountain, tree, broccoli.
Sue swallowed the large green _______. pill, broccoli.

- n元模型: 根据n-1个词的历史，预测下面的词是哪个词？

n越大，历史信息越多，模型越准确

n的选择

- n 较大时
  – 提供了更多的语境信息，语境更具区别性
  – 参数个数多、计算代价大、训练语料需要多、参数估计不可靠。
- n 较小时
  – 语境信息少，不具区别性
  – 但是，参数个数少、计算代价小、训练语料无需太多、参数估计可靠。

马尔可夫模型

- n元模型 = n-1阶马尔可夫过程
- n元模型把句子视作马尔可夫过程的产物
  - 始于标志句子开始的词$\langle bos\rangle$
  - 逐步生成句子中的词，直到生成标志句子结束的词$\langle eos\rangle$

$$\langle bos\rangle \xrightarrow{P(*|\langle bos\rangle)} John \xrightarrow{P(*|John)} read \xrightarrow{P(*|read)} a \xrightarrow{P(*|a)} book \xrightarrow{P(*|book)} \langle eos\rangle$$

$$P(John\ read\ a\ book)$$
$$= P(John|\langle bos\rangle) \times P(read|John) \times P(a|read) \times P(book|a)$$
$$\times P(\langle eos\rangle|book)$$

如何建立n元模型

- 数据准备:
  - 确定训练语料
  - 对语料进行词例化(tokenization) 或切分
  - 句子边界标记，增加两个特殊的词<bos>和<eos>
    - I eat . → <bos> I eat . <eos>
    - I sleep . → <bos> I sleep . <eos>
    -
- 参数估计
  - 利用训练语料，估计模型参数

相对频率法

- 令 $c(w_1w_2 \dots w_n)$表示n元组 $w_1w_2 \dots w_n$在训练语料中出现的次数。则：

$$P(w_n|w_1 \dots w_{n-1}) = \frac{c(w_1w_2 \dots w_n)}{c(w_1w_2 \dots w_{n-1})}$$

- 训练语料

  <bos> John read Moby Dick <eos>

  <bos> Mary read a different book <eos>

  <bos> She read a book by Cher <eos>

$$P(John|\langle bos\rangle) = \frac{c(\langle bos\rangle John)}{c(\langle bos\rangle)} = \frac{1}{3}$$

$$P(a|read) = \frac{c(read\ a)}{c(read)} = \frac{2}{3}$$

$$P(\langle eos\rangle|book) = \frac{c(book\langle eos\rangle)}{c(book)} = \frac{1}{2}$$

$$P(book|a) = \frac{c(a\ book)}{c(a)} = \frac{1}{2}$$

$$P(read|John) = \frac{c(John\ read)}{c(John)} = \frac{1}{1}$$

……

最大似然估计

- $P(T;\Theta)$是训练语料的概率，假设句子和句子互相独立

$$P(T;\Theta) = \prod_{i=1}^{n} P(s_i;\Theta)$$

- 原则：选择使训练样本似然值(概率)最大的参数

$$\Theta_{ML} = \operatorname*{argmax}_{\Theta} P(T;\Theta)$$

  $P(T;\Theta)$是训练语料的概率，Θ代表所有参数(条件概率)

- 该优化问题具有解析解

  参数的最大似然估计值=参数的相对频率估计值

$$\Theta_{ML} = \Theta_{RF}$$

计算句子的概率

- 计算句子*John read a book*的概率

$$
\begin{aligned}
&P(John\ read\ a\ book) \\
&= P(John|\langle bos\rangle) \times P(read|John) \times P(a|read) \times P(book|a) \\
&\times P(\langle eos\rangle|book) \\
&= \frac{1}{3} \times 1 \times \frac{2}{3} \times \frac{1}{2} \times \frac{1}{2} \\
&= 0.06
\end{aligned}
$$

- 避免运算下溢(underflow): 计算对数概率

$$
\begin{aligned}
&\ln P(John\ read\ a\ book) \\
&= \ln P(John|\langle bos\rangle) + \ln P(read|John) + \ln P(a|read) + \ln P(book|a) \\
&+ \ln P(\langle eos\rangle|book) \\
&= -2.8902
\end{aligned}
$$

数据稀疏(Data Sparseness)

- 考虑计算句子*Cher read a book*的概率。

  $c(Cher\ read) = 0$
  → $P(read|Cher)=0$
  → $P(Cher\ read\ a\ book) = 0$ (有问题)

- 若n元组在训练语料中没有出现，则该n元组的概率必定是0
- 最大似然估计法(MLE)给训练样本中未观察到的事件赋 0 概率
- 由于训练样本不足而导致所估计的分布不可靠的问题，称为**数据稀疏问题**
- 解决的办法:
  扩大训练语料的规模?

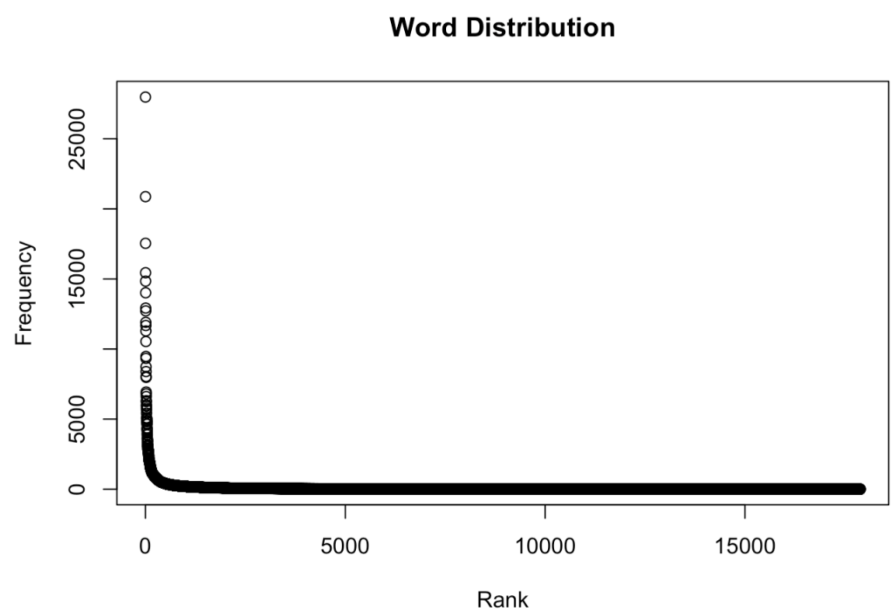
Zipf 定律

- Zipf **定律**: **词频**和**序号**之间的关系

  针对给定的语料库，若某个词w的词频是f，且该词在词频表中的序号为r，则

  $f \times r = k$ 且 k (大致)是一个常数

Word	Frequency	Rank
the	27976	1
and	20869	2
a	17540	3
to	15442	4
of	14840	5
…	…	…



Zipf 定律

- Tom Sawyer (by Mark Twain)
  - word tokens: 71,370
  - word types: 8,018

Word Frequency	Frequency of Frequency
1	3993
2	1292
3	664
4	410
5	243
6	199
7	172
8	131
9	82
10	91
11-50	540
51-100	99
> 100	102

- Frequency of Frequency
  频率是f的词有几个

- 大部分词是低频词
  3993 个词(约50%)仅出现了一次

- 常用词极为常用
  前100个高频词占了约51%的文本篇幅

Zipf 定律

- 语言中只有很少的常用词
- 语言中大部分词都是低频词
- 词的分布是长尾分布，n元组分布亦是如此

- 大多数词(n元组)在语料中的出现是稀疏的
- 语料库可以提供少量常用词(n元组)的可靠样本
- 语料库规模扩大，主要是高频词词例的增加
- 扩大语料规模不能从根本上解决稀疏问题

数据稀疏

- Bahl et al. 1983
  - 用150 万词的训练语料训练三元模型
  - 测试语料(同样来源)中有23%的三元组没有在训练语料中出现过

- 由于数据稀疏，MLE估计值不是理想的参数估计值

- 解决办法: 平滑(smoothing)
  - 把在训练样本中出现过的事件的概率适当减小
  - 把减小得到的概率质量分配给训练语料中没有出现过的事件
  - 这个过程有时也称作减值(discounting)法

加法平滑

- 不同的减值策略 $\Rightarrow$ 不同的平滑方法
- 最简单的平滑方法是加法平滑

- 加1平滑：规定n元组比真实出现次数多一次

$$new\_count(n\text{−}gram) = old\_count(n\text{−}gram) + 1$$

- 没有出现过的n元组的概率不再是0，而是一个较小的概率值，实现了概率质量的重新分配

加法平滑

$$p_{MLE}(w_n|w_1 \dots w_{n-1}) = \frac{c(w_1 w_2 \dots w_n)}{c(w_1 w_2 \dots w_{n-1})}$$

$$\Downarrow$$

$$p_{+1}(w_n|w_1 \dots w_{n-1}) = \frac{c(w_1 w_2 \dots w_n) + 1}{c(w_1 w_2 \dots w_{n-1}) + |V|}$$

V 代表词表，$|V|$代表词表中词的数量

加法平滑

2nd word

1st word	*I*	*want*	*to*	*eat*	*Chinese*	*food*	*lunch*	…	**Total (N)**
I	8	1087	0	13	0	0	0		3437
want	3	0	786	0	6	8	6		1215
to	3	0	10	860	3	0	12		3256
eat	0	0	2	0	19	2	52		938
Chinese	2	0	0	0	0	120	1		213
food	19	0	17	0	0	0	0		1506
lunch	4	0	0	0	0	1	0		459
…									

未平滑的 bigram 频次

	I	*want*	*to*	*eat*	*Chinese*	*food*	*lunch*	…	**Total**
I	.0023 (8/3437)	.32	0	.0038 (13/3437)	0	0	0		1
want	.0025	0	.65	0	.0049	.0066	.0049		1
to	.00092	0	.0031	.26	.00092	0	.0037		1
eat	0	0	.0021	0	.020	.0021	.055		1
Chinese	.0094	0	0	0	0	.56	.0047		1
food	.013	0	.011	0	0	0	0		1
lunch	.0087	0	0	0	0	.0022	0		1
…									

未平滑的 bigram 概率 $p(w_2|w_1)$

加法平滑

平滑后的 bigram 频次

	I	*want*	*to*	*eat*	*Chinese*	*food*	*lunch*	…	**Total (N+V)**
I	~~8~~ 9	~~1087~~ 1088	1	14	1	1	1		~~3437~~ 5053
want	~~3~~ 4	1	787	1	7	9	7		2831
to	4	1	11	861	4	1	13		4872
eat	1	1	23	1	20	3	53		2554
Chinese	3	1	1	1	1	121	2		1829
food	20	1	18	1	1	1	1		3122
lunch	5	1	1	1	1	2	1		2075

	I	*want*	*to*	*eat*	*Chinese*	*food*	*lunch*	…	**Total**
I	.0018 (9/5053)	.22	.0002	.0028 (14/5053)	.0002	.0002	.0002		1
want	.0014	.00035	.28	.00035	.0025	.0032	.0025		1
to	.00082	.00021	.0023	.18	.00082	.00021	.0027		1
eat	.00039	.00039	.0012	.00039	.0078	.0012	.021		1
Chinese	.0016	.00055	.00055	.00055	.00055	.066	.0011		1
food	.0064	.00032	.0058	.00032	.00032	.00032	.00032		1
lunch	.0024	.00048	.00048	.00048	.00048	.0022	.00048		1

平滑后的 bigram概率 $p(w_1|w_2)$

加法平滑

- 训练语料中未出现的n元组的概率不再为0，而是一个大于0的较小的概率值。

- 但由于训练语料中未出现n元组数量太多，平滑后，所有未出现的n元组占据了整个概率分布中的一个很大的比例。因此，在NLP中，加1平滑给训练语料中没有出现过的n元组分配了太多的概率空间。

- AP 语料数据 (Church and Gale, 1991)
  - 语料共含有 2200 0000 个二元组 (token), 涉及320 6756二元组(type)
  - 语料中共出现了 27 3266 个词(type) (共有 746 7430 6756个可能的二元组(type))
  - 746 7110 0000 二元组在语料中没有出现
  - 利用加1平滑，每个未出现过二元组平均出现 0.000295次
  - 调整概率也就是调整频次

训练语料中的频次

Add-one 平滑后的频次

f_{MLE}	$f_{add\text{-}one}$
0	**0.000295**
1	0.000589
2	0.000884
3	0.001178
4	0.001473
5	0.001767

太大了

太小了

- 所有未出现过的二元组的概率分布 =

  (746 7110 0000 × 0.000295) / 2200 0000 ~**99.96** !!!!

加法平滑

- 加δ平滑

$$p_{MLE}(w_n|w_1 \dots w_{n-1}) = \frac{c(w_1 w_2 \dots w_n)}{c(w_1 w_2 \dots w_{n-1})}$$

$$\Downarrow$$

$$p_{+1}(w_n|w_1 \dots w_{n-1}) = \frac{c(w_1 w_2 \dots w_n) + \delta}{c(w_1 w_2 \dots w_{n-1}) + \delta|V|}$$

其中，$0 < \delta < 1$

- 加法平滑是简单平滑法，折减的概率质量平均分配

简单平滑

- 设历史是H，语料中出现过HA, HB, HC，但没有出现过HD, HE，按照MLE估计

$$p(A|H) + p(B|H) + p(C|H) = 1$$
$$p(D|\mathrm{H}) = p(E|H) = 0$$

- 折扣，则

$$\hat{p}(A|H) + \hat{p}(B|H) + \hat{p}(C|H) = \delta$$

- **平均分配**

$$\hat{p}(D|\mathrm{H}) = \hat{p}(E|H) = \frac{1-\delta}{2}$$

组合平滑方法

- 不同的平滑方法 ⇒ 如何折扣？如何分配？
- 简单平滑 平均分配折减概率质量
- 平均分配折减出的概率质量，是否合理？
  - 例如，下面三个bigram均未出现
    - journal of $p_{MLE}(\text{of}|\text{journal}) = 0$
    - journal from $p_{MLE}(\text{from}|\text{journal}) = 0$
    - journal never $p_{MLE}(\text{never}|\text{journal}) = 0$
- "journal of" 更常见，概率应该更大
  - "of" 频率高于"from" & "never"
  - unigram 概率应为$p(of) > p(from) > p(never)$
- 组合平滑方法 利用低阶模型指导概率质量分配

组合平滑方法

- 分配策略：
  （1）回退策略
  （2）插值策略
- 回退策略：只有D, E参加分配

$$\hat{p}(D|H) = \frac{p(D)}{P(D) + p(E)} \cdot (1 - \delta)$$

$$\hat{p}(E|H) = \frac{p(E)}{P(D) + p(E)} \cdot (1 - \delta)$$

- 可以写成

$$\hat{p}(w|H) = \begin{cases} \delta \cdot p(w|H), & \text{if } w \in \{A, B, C\} \\ (1 - \delta) \cdot \dfrac{p(w)}{\sum_{v \in \{D,E\}} p(v)}, & \text{if } w \in \{D, E\} \end{cases}$$

组合平滑方法

- 插值平滑方法，A, B, C, D, E都参加分配

$$\hat{p}(A|H) = \hat{p}(A|H) + p(A) \cdot (1 - \delta)$$
$$\hat{p}(B|H) = \hat{p}(B|H) + p(B) \cdot (1 - \delta)$$
$$\hat{p}(C|H) = \hat{p}(C|H) + p(C) \cdot (1 - \delta)$$

$$\hat{p}(D|H) = 0 + p(D) \cdot (1 - \delta)$$
$$\hat{p}(E|H) = 0 + p(E) \cdot (1 - \delta)$$

- 即 $\hat{p}(w|H) = \hat{p}(w|H) + (1 - \delta) \cdot p(w)$

绝对减值法导引

- 如何折减？折减绝对频次

$$\hat{p}(A|H) = \frac{c(HA) - \delta}{c(H)}$$

$$\hat{p}(B|H) = \frac{c(HB) - \delta}{c(H)}$$

$$\hat{p}(C|H) = \frac{c(HC) - \delta}{c(H)}$$

- 总计折扣出的概率质量$3\delta/c(H)$，记作$1 - \lambda_H$
- 按照插值策略进行分配

$$\hat{p}(w|H) = \hat{p}(w|H) + \frac{3\delta}{c(H)} \cdot p(w) = \hat{p}(w|H) + (1 - \lambda_H) \cdot p(w)$$

绝对减值法

- 对所有出现过的n元组折扣固定的绝对频次δ（$0 < \delta < 1$），若$c\left(w_{i-n+1}^{i}\right) > 0$

$$c\left(w_{i-n+1}^{i}\right) - \delta$$

- 给定历史w_{i-n+1}^{i-1}，可供折扣频次的n元组个数

$$N_{1+}\left(w_{i-n+1}^{i-1} *\right) = \left|\left\{w_i : c\left(w_{i-n+1}^{i-1} w_i\right) > 0\right\}\right|$$

- 给定历史w_{i-n+1}^{i-1}，折扣出的总频次

$$\delta \cdot N_{1+}\left(w_{i-n+1}^{i-1} *\right)$$

- 总计折扣出的概率质量

$$\frac{\delta}{\sum_{w_i} c\left(w_{i-n+1}^{i}\right)} N_{1+}\left(w_{i-n+1}^{i-1} *\right) = 1 - \lambda_{w_{i-n+1}^{i-1}}$$

绝对减值法

- 将折扣出的概率质量根据低阶模型分配
- 绝对减值法是插值模型，递归定义如下

$$p_{abs}(w_i|w_{i-n+1}^{i-1}) = \frac{\max\{c(w_{i-n+1}^{i}) - \delta, 0\}}{\sum_{w_i} c(w_{i-n+1}^{i})} + \left(1 - \lambda_{w_{i-n+1}^{i-1}}\right) p_{abs}(w_i|w_{i-n+2}^{i-1})$$

- 如何选择δ?
  - $\delta = 0.75$
  - $\delta_{r=1} = 0.5$；$\delta_{r\geq 2} = 0.75$
  - $\delta = \frac{N_1}{N_1 + 2N_2}$
- 模型递归终止于一元模型

Kneser-Ney平滑导引

- 低阶模型决定概率重分配的比重，合理的低阶模型很重要
- KN平滑是对绝对减值法的改进，修改了低阶模型的定义

- 按照低阶模型分配不一定好，因为$p(w)$大，就给$p(w|H)$多分配一些，未必合理
  - 因为w出现在H后面的可能性有可能非常小
  - 最好根据w出现在H后面可能性进行分配

- 根据w前驱词种数判断其出现在陌生环境中的可能性

Kneser–Ney平滑

- 词的前驱词数量各异

  San Francisco　　　Francisco只能出现San之后

  语料中大量出现 → $p(\text{Francisco})$

  I can't see without my reading ________

$$p(\text{Francisco}) > p(\text{glasses})$$

- 若采用常规的低阶模型

  $p_{abs}(\text{Francisco}|\text{reading})$、$p_{abs}(\text{Francisco}|\text{wear})$…都会被分配较大的概率

- 使用常规低阶模型，导致

$$\sum_{w_{i-1}} p_{abs}(w_{i-1}w_i) \neq p(w_i)$$

Kneser–Ney平滑

- 按照w_i的前驱词数量分配概率质量，可避免此问题
  glasses的前驱词种数大于Francisco前驱词的种数
  buy glasses, wear glasses,…
- 统计词w_i的前驱词种数$\left|\{w_{i-1}: c(w_{i-1}w_i) > 0\}\right|$
- 接续概率(continuation probability)

$$p_{cnt}(w_i) = \frac{\left|\{w_{i-1}: c(w_{i-1}w_i) > 0\}\right|}{\sum_{w_j}\left|\{w_{j-1}: c\left(w_{j-1}w_j\right) > 0\}\right|}$$

- Kneser-Ney平滑(二元模型)

$$p_{KN}(w_i|w_{i-1}) = \frac{\max\{c(w_{i-1}w_i) - \delta, 0\}}{\sum_{w'} c(w_{i-1}w')} + \left(1 - \lambda_{w_{i-1}}\right)p_{cnt}(w_i)$$

Kneser–Ney平滑

- 修正频次计算

$$c_{KN}\left(w_{i-n+1}^{i}\right)=\begin{cases}c\left(w_{i-n+1}^{i}\right), & if\ n=N\\ \left|v: c\left(vw_{i-n+1}^{i}\right)>0\right|, & if\ n<N\end{cases}$$

- Kneser-Ney模型(N元模型)

$$p_{KN}\left(w_i \middle| w_{i-n+1}^{i-1}\right)=\frac{\max\{c_{KN}\left(w_{i-n+1}^{i}\right)-\delta,0\}}{\sum_{w_i} c_{KN}\left(w_{i-n+1}^{i}\right)}+\left(1-\lambda_{w_{i-n+1}^{i-1}}\right)p_{KN}\left(w_i \middle| w_{i-n+2}^{i-1}\right)$$

- 递归终止于一元模型

熵(Entropy)

- 什么是熵？

  设X是取有限个值的随机变量，若其概率分布为$p(x)$， 且$x \in \mathcal{X}$，则X的熵定义为

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_a p(x)$$

- 规定$0 \log_a 0$=0
- 通常a=2，此时熵的单位为比特。

- 令X代表硬币抛掷结果，正面(H)和反面(T)朝上的概率均为$\frac{1}{2}$

$$H(X) = -(p(H) \log_2 p(H) + p(T) \log_2 p(T))$$
$$= -(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2}) = 1$$

熵

- 熵的基本性质:
  $H(X) \geq 0$, 等号表明确定场(无随机性)的熵最小。
  $H(X) \leq \log |\mathcal{X}|$, 等号表明等概场的熵最大。

- 熵描述了随机变量的不确定性
- 概率越小的事件蕴含越大的信息量
  - 人咬狗 vs. 狗咬人
- 熵描述了随机变量的平均信息量

联合熵和条件熵

- 设X、Y是两个离散型随机变量，它们的联合分布为$p(x,y)$，则X 、Y的联合熵定义为

$$H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(x,y)$$

- 设X、Y是两个离散型随机变量，它们的联合分布为$p(x,y)$，则给定X时Y的条件熵定义为

$$\begin{aligned} H(Y|X) &= \sum_{x\in\mathcal{X}} p(x)H(Y|X=x) \\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(y|x) \end{aligned}$$

- 链式规则　$H(X,Y) = H(Y|X) + H(X)$

相对熵

- 设$p(x)$ 是随机变量X的真实分布密度，$q(x)$是通过统计手段得到的X的近似分布，则二者间相对熵定义为

$$D(p \parallel q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}$$

$$= -\sum_{x \in \mathcal{X}} p(x) \log q(x) - \left(-\sum_{x \in \mathcal{X}} p(x) \log p(x)) \right)$$

- 相对熵也称作
  - Kullback-Leibler 发散度、KL发散度
  - Kullback-Leibler 距离、KL距离
- 相对熵描述同一个随机变量的不同分布的差异程度
- 相对熵描述了因为错用分布密度而增加的信息量(不确定性)

交叉熵

- 设随机变量X的分布密度为$p(x)$，$q(x)$是通过统计手段得到的X的近似分布，则随机变量X的交叉熵定义为

$$H(X,q) = -\sum_{x \in \mathcal{X}} p(x) \log q(x)$$

- 与相对熵的关系

$$D(p \parallel q) = H(X,q) - H(X)$$

- 若存在X的两个近似分布$q_1(x)$、$q_2(x)$
  $H(X,q_1) < H(X,q_2)$，q_1是更好的近似分布

语言模型的评价—交叉熵

- 令T为测试语料

$$T = w_1 w_2 \dots w_{N^T}$$

- 语言模型的交叉熵

$$H_{lm}(T) = -\frac{1}{N^T} \log p(T)$$

- 计算$P(T)$

$$P(T) = \prod_{i=1}^{N^T} p(w_i | w_{i-n+1} \dots w_{i-1})$$

- 交叉熵越小，语言模型质量越好

语言模型的评价—交叉熵

- w在T中的经验分布

$$\tilde{p}(w) = \frac{c(w)}{N^T}$$

- 交叉熵衡量一元模型与测试语料经验分布的差异

$$H_{lm}(T) = -\frac{1}{N^T}\log p(T) = -\sum_{w \in V_T} \tilde{p}(w)\log p(w)$$

一元模型

- $w_1 w_2 \ldots w_n$在T中的经验分布

$$\tilde{p}(w_1 \ldots w_{n-1} w_n) = \frac{c(w_1 w_2 \ldots w_n)}{N^T}$$

- 交叉熵衡量n元模型与测试语料经验分布的差异

$$H_{lm}(T) = -\frac{1}{N^T}\log p(T)$$

$$= -\sum_{w_1 \ldots w_n \in V^n} \tilde{p}(w_1 w_2 \ldots w_n)\log p(w_n | w_1 \ldots w_{n-1})$$

n元模型

困惑度(perplexity)

- 语言模型的评价也可使用困惑度：

$$PP_{lm}(T) = 2^{H_{lm}(T)}$$

$$= \sqrt[N_T]{\frac{1}{\prod_{i=1}^{N_T} p(w_i|w_{i-n+1} \dots w_{i-1})}}$$

- 根据n元模型，正确采样w_i的平均(几何平均)采样次数

- 与交叉熵的度量结果一致
  交叉熵 9.9 → 9.1
  困惑度 950 → 540