第 4 章　统计语言模型

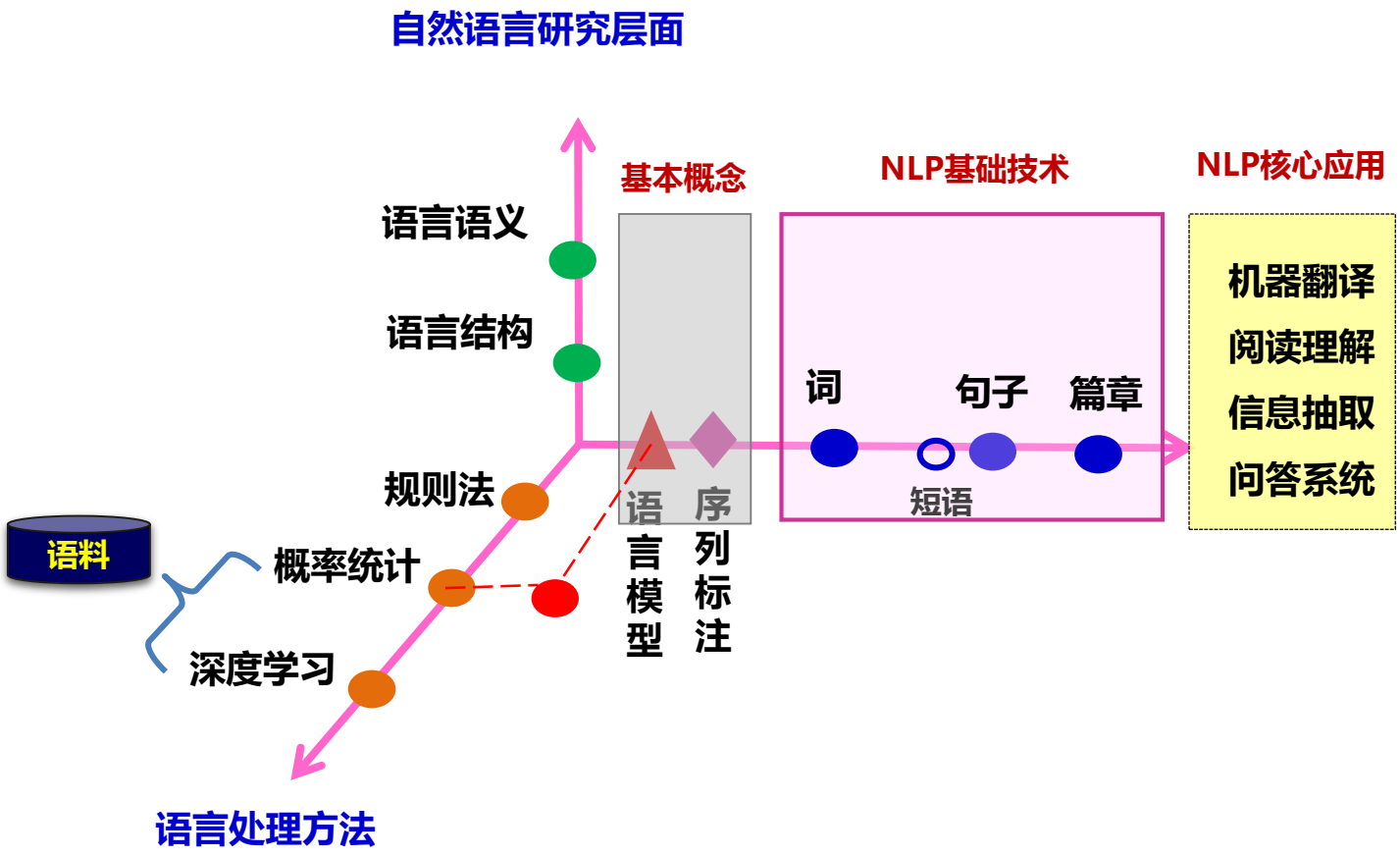
中科院信息工程研究所第二研究室

胡玥

huyue@iie.ac.cn

自然语言处理课程内容及安排

✧ 课程内容：



内 容 提 要

4.1 语言模型基本概念

问题引入：

下表中，给定拼音串，如何确定对应的文字？

拼音串（无声调）	ni xian zai zai gan shen mo
候选字串	你 线 在 再 干 什 么
	你 现 在 在 干 什 么
	尼 先 在 在 感 什 么
	……
候选词串	你 现在 在 感什么
	你 现在 在 干什么
	你 先在 再 干什么
	……
正确文字串	你现在在干什么

4.1 语言模型基本概念

语言模型提出

弗莱德里克·贾里尼克（美国工程院院士）

在“基于统计的语音识别的框架”中提出了.

用数学的方法描述语言规律（语言模型）

即，用句子S=w1,w2,...,wn 的**概率** p(S) 刻画句子的合理性。（而不需进行语言学分析处理）。

统计自然语言处理的基础模型

对语句合理性判断：

规则法：判断是否合乎语法、语义（定性分析）

统计法： 通过可能性（概率）的大小来判断（定量计算）

主要源自解决语音识别问题

4.1 语言模型基本概念

语言模型思想

用句子S=w1,w2,...,wn 的 概率 p(S) 来定量的刻画句子

句子$S=w_1,w_2,\ldots,w_n$ 的概率 p(S)：

自然语言为上下文相关的信息传递方式

语句 $s = w_1\ w_2\ \ldots\ w_n$ 的 概率p(S)定义为 ：

$p(S)=p(w_1)p(w_2|w_1)\ldots p(w_n|w_1,\ldots,w_{n-1})$

$$= \prod_{i=1}^{n} p(w_i \mid w_1 \cdots\cdots w_{i-1})$$

其中：当 i=1 时，$p(w_1|w_0) = p(w_1)$

4.1 语言模型基本概念

语言模型

$$p(S) = \prod_{i=1}^{n} p(w_i \mid w_1 \cdots\cdots w_{i-1})$$

输入： 句子 S

输出： 句子概率p(S)

参数： $p(w_i|w_1,\ldots,w_{i-1})$

说明：

(1) w_i 可以是字、词、短语或词类等等，称为统计基元。通常以“词”代之。

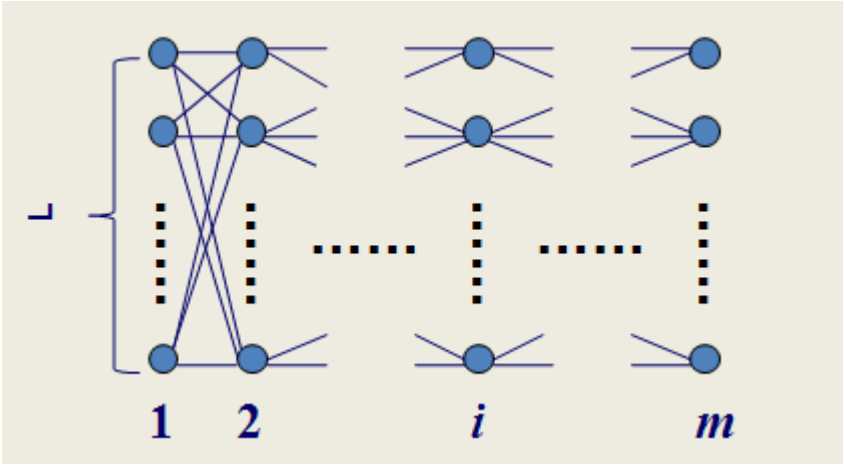
(2) w_i 的概率由 $w_1, \ldots, w_{i-1}$ 决定，由特定的一组 $w_1, \ldots, w_{i-1}$ 构成的一个序列，称为 w_i 的历史（history）。

4.1 语言模型基本概念

原始定义存在的问题：

第 i （ i >1 ）个统计基元，历史基元的个数为 i-1，如果共有 L 个不同的基元(如词汇表)，i 基元就有 L^{i-1} 种不同的历史情况。我们必须考虑在所有的 L^{i-1} 种不同历史情况下产生第 i 个基元的概率。模型中有L^m 个自由参数 $p(w_m|w_1 \ldots w_{m-1})$。

$p(w_i|w_1,\ldots,w_{i-1})$



如果 L=5000, m = 3, 自由参数的数目为 1250 亿！

一个汉语句子平均有22个词

4.1 语言模型基本概念

问题解决方法

减少历史基元的个数，马尔可夫方法：假设任意一个词 w_i出现的概率只与它前面的w_{i-1} 有关，问题得以简化

$$p(S)=p(w_1)p(w_2|w_1)\ldots p(w_n|w_1,\ldots,w_{n-1})$$

$$p(s) = p(w_1)\times p(w_2|w_1)\times p(w_3|w_2)\times \ldots \times p(w_n|w_{n-1})$$

二元模型

4.1 语言模型基本概念

n 元文法(n-gram)

n 元文法(n-gram)：一个词由前面的 n-1 个词决定

- **当 *n*=1 时，即出现在第 i 位上的基元 w_i 独立于历史。**

$$p(w_i|w_1,\ldots,w_{i-1})=p(w_i)$$

一元文法也被写为 uni-gram **或** monogram ；

- **当 *n*=2 时, 2-gram** (bi-gram) **被称为1阶马尔可夫链；**

$$p(w_i|w_1,\ldots,w_{i-1}) = p(w_i|w_{i-1})$$

- **当 *n*=3 时, 3-gram**(tri-gram)**被称为2阶马尔可夫链，**

$$p(w_i|w_1,\ldots,w_{i-1}) = p(w_i|w_{i-2}, w_{i-1})$$

- **依次类推**

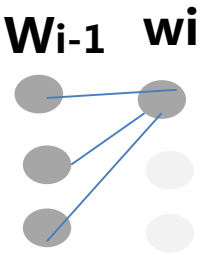
4.1 语言模型基本概念

◆ 理论上讲，N 越大越好

句子S=w1,w2,...,wn 的 概率 p(S)

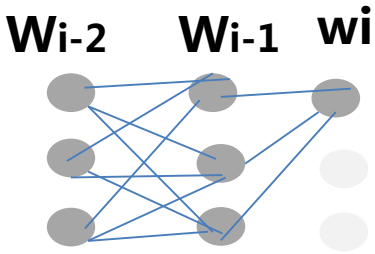
$$p(S) = \prod_{i=1}^{n} p(w_i \mid w_1 \cdots\cdots w_{i-1})$$

◆ 但 N 越大，需要估计的参数越多



$p(w_i|w_{i-1})$

参数：3



$p(w_i|w_{i-2}w_{i-1})$

参数：9

◆ 经验值：

3，tri-gram用的最多 ； 4，four-gram需要太多的参数，少用。

高阶模型也无法覆盖所有的语言现象

4.1 语言模型基本概念

例： **给定句子**：John read a book **求 概率**

解： **增加标记**：<BOS> John read a book <EOS>

基于**1元**文法的概率为：

$$p(\text{John read a book}) = p(\text{John}) \times p(\text{read}) \times p(\text{a}) \times P(\text{book})$$

基于**2元**文法的概率为：

$$p(\text{John read a book}) = p(\text{John}|\text{<BOS>}) \times p(\text{read}|\text{John}) \times p(\text{a}|\text{read}) \times$$
$$p(\text{book}|\text{a}) \times p(\text{<EOS>}|\text{book})$$

问题：如何获得 *n* 元语法模型中的各概率值（参数）？

内 容 提 要

4.2.1 参数估计

参数估计（模型训练）：获得模型中所有的条件概率（模型参数）

1. 训练语料：

- **已知语料**
- **训练语料应尽量和应用领域一致**
- **语料尽量足够大**
- **训练前应预处理**

语言模型对于训练文本的类型、主题和风格等都十分敏感

4.2.1 参数估计

2. 参数学习的方法

对于 *n*-gram，参数 $p(w_i \mid w_{i-n+1}^{i-1})$ 可由最大似然估计求得:

$$p(w_i \mid w_{i-n+1}^{i-1}) = f(w_i \mid w_{i-n+1}^{i-1}) = \frac{\sum_{w_i} c(w_{i-n+1}^{i})}{\sum_{w_i} c(w_{i-n+1}^{i-1})}$$

其中：

$\sum_{w_i} c(w_{i-n+1}^{i-1})$ 是历史串 w_{i-n+1}^{i-1} 在给定语料中出现的次数

$\sum_{w_i} c(w_{i-n+1}^{i})$, 为 w_{i-n+1}^{i-1} 与 w_i 同现的次数。

最大似然估计(*maximum likelihood Evaluation*, MLE)

4.2.1 参数估计

例如：给定训练语料：

“John read Moby Dick”

“Mary read a different book”

“She read a book by Cher”

如何 训练 2 元文法

$$p(\text{John read a book}) = p(\text{John}|<\text{BOS}>) \times p(\text{read}|\text{John}) \times p(\text{a}|\text{read}) \times p(\text{book}|\text{a}) \times p(<\text{EOS}>|\text{book})$$

参数

4.2.1 参数估计

解： **参数：p(John|<BOS>)，p(read|John)，p(a|read)**

p(book|a)，p(<EOS>|book)

$$p(John \mid <BOS>) = \frac{c(<BOS> \; John)}{\sum_w c(<BOS> \; w)} = \frac{1}{3} \qquad p(read \mid John) = \frac{c(John \quad read)}{\sum_w c(John \quad w)} = \frac{1}{1}$$

$$p(a \mid read) = \frac{c(read \quad a)}{\sum_w c(read \quad w)} = \frac{2}{3} \qquad p(book \mid a) = \frac{c(a \quad book)}{\sum_w c(a \quad w)} = \frac{1}{2}$$

$$p(<EOS> \mid book) = \frac{c(book \quad <EOS>)}{\sum_w c(book \quad w)} = \frac{1}{2}$$

<BOS>John read Moby Dick<EOS>

<BOS>Mary read a different book<EOS>

<BOS>She read a book by Cher<EOS>

4.2.1 参数估计

句子 *John read a book* **的概率**

基于**2元**文法的概率为：

$$p(\text{John read a book}) = p(\text{John}|\text{<BOS>}) \times p(\text{read}|\text{John}) \times p(\text{a}|\text{read}) \times p(\text{book}|\text{a}) \times p(\text{<EOS>}|\text{book})$$

$$p(John\ read\ a\ book) = \frac{1}{3} \times 1 \times \frac{2}{3} \times \frac{1}{2} \times \frac{1}{2} \approx 0.06$$

语言模型的用途

决定哪一个词序列的可能性更大

已知若干个词，预测下一个词….

4.2.1 参数估计

问题： 如求，$p(Cher\ read\ a\ book) = ?$

$$=p(Cher|<BOS>)\times p(read|Cher)\times p(a|read)\times$$
$$p(book|a)\times p(<EOS>|book)$$

$$p(Cher\ |<BOS>) = \frac{c(<BOS>\ Cher)}{\sum_w c(<BOS>\ w)} = \frac{0}{3}$$

$$p(read\ |\ Cher) = \frac{c(Cher\quad read)}{\sum_w c(Cher\quad w)} = \frac{0}{1}$$

$$p(read\ |\ John) = \frac{c(John\quad read)}{\sum_w c(John\quad w)} = \frac{1}{1}$$

于是，$p(Cher\ read\ a\ book) = 0$

数据匮乏(稀疏) (*Sparse Data*) 引起零概率问题

<BOS>John read Moby Dick<EOS>

<BOS>Mary read a different book<EOS>

<BOS>She read a book by Cher<EOS>

内 容 提 要

4.2.2 数据平滑

1. 数据平滑的基本思想：

调整最大似然估计的概率值,使零概率增值，使非零概率下调，

“劫富济贫”，消除零概率，改进模型的整体正确率。

- 基本目标：测试样本的语言模型困惑度越小越好。
- 基本约束：$\sum_{w_i} p(w_i \mid w_1, w_2, \cdots, w_{i-1}) = 1$

4.2.2 数据平滑

2. 数据平滑方法：

- **加1法**(Additive smoothing)
- **减值法/折扣法** (Discounting)
  - 1) Good-Turing
  - 2) Back-off (Katz)
  - 3) 绝对减值(H. Ney)
  - 4) 线性减值
- **删除减值法：低阶代替高阶**

4.2.2 数据平滑

◆ 加1法 (Additive smoothing)

基本思想: 每一种情况出现的次数加1。

如，对于2-gram 有：

$$p(w_i \mid w_{i-1}) = \frac{1 + c(w_{i-1}w_i)}{\sum_{w_i}[1 + c(w_{i-1}w_i)]} = \frac{1 + c(w_{i-1}w_i)}{|V| + \sum_{w_i} c(w_{i-1}w_i)}$$

其中，V 为被考虑语料的词汇量（全部可能的基元数）

4.2.2 数据平滑

问题回顾：

如求，$p(Cher\ read\ a\ book) = ?$

$$=p(Cher|<BOS>)\times p(read|Cher)\times p(a|read)\times p(book|a)\times p(<EOS>|book)$$

$$p(Cher|<BOS>)=\frac{c(<BOS>\ Cher)}{\sum_w c(<BOS>\ w)}=\frac{0}{3}$$

$$p(read|Cher)=\frac{c(Cher\ \ read)}{\sum_w c(Cher\ \ w)}=\frac{0}{1}$$

p(Cher read a book) = 0

原来：

p(Cher|<BOS>) = 0/3

p(read|Cher) = 0/1

$p(a|read) = 2/3$

$p(book|a) = 1/2$

$p(<EOS>|book)= 1/2$

平滑处理：

<BOS>John read Moby Dick<EOS>

<BOS>Mary read a different book<EOS>

<BOS>She read a book by Cher<EOS>

4.2.2 数据平滑

原来：

***p*(*Cher*|<*BOS*>) = 0/3**

***p*(*read*|*Cher*) = 0/1**

p(*a*|*read*) = 2/3

p(*book*|*a*) = 1/2

p(<*EOS*>|*book*)= 1/2

平滑以后：

***p*(*Cher*|<*BOS*>) = (0+1)/(11+3) = 1/14**

***p*(*read*|*Cher*) = (0+1)/(11+1) = 1/12**

***p*(*a*|*read*) = (1+2)/(11+3) = 3/14**

***p*(*book*|*a*) = (1+1)/(11+2) = 2/13**

***p*(<*EOS*>|*book*)= (1+1)/(11+2) = 2/13**

词汇量：|V| = 11

***p*(*Cher read a book*)**

$$= p(Cher|<BOS>) \times p(read|Cher) \times p(a|read) \times p(book|a) \times p(<EOS>|book)$$

$$= \frac{1}{14} \times \frac{1}{12} \times \frac{3}{14} \times \frac{2}{13} \times \frac{2}{13} \approx 0.00003$$

<BOS>John read Moby Dick<EOS>

<BOS>Mary read a different book<EOS>

<BOS>She read a book by Cher<EOS>

4.2.2 数据平滑

同理，对于句子*John read a book* 数据平滑后：

平滑后：

$p(John|<BOS>) = 2/14,$

$p(read|John) = 2/12,$

$p(a|read) = 3/14,$

$p(book|a) = 2/13,$

$p(<EOS>|book) = 2/13$

原来：

$p(John|<BOS>) = 1/3,$

$p(read|John) = 1/1,$

$p(a|read) = 2/3,$

$p(book|a) = 1/2,$

$p(<EOS>|book) = 1/2$

$$p(John\ read\ a\ book)$$

$$= p(John|<BOS>) \times p(read|John) \times p(a|read) \times p(book|a) \times p(<EOS>|book)$$

$$= \frac{2}{14} \times \frac{2}{12} \times \frac{3}{14} \times \frac{2}{13} \times \frac{2}{13} \approx 0.0001$$

<BOS>John read Moby Dick<EOS>

<BOS>Mary read a different book<EOS>

<BOS>She read a book by Cher<EOS>

4.2.2 数据平滑

- **减值法/折扣法 (Discounting)**

基本思想：修改训练样本中事件的实际计数，使样本中(实际出现的)不同事件的概率之和小于1，剩余的概率量分配给未见概率。

1) Good-Turing　　2) Back-off (Katz)

3) 绝对减值(H. Ney)　　4) 线性减值

4.2.2 数据平滑

- **删除插值法(Deleted interpolation)**

基本思想：用低阶语法估计高阶语法，即当 3-gram 的值不能从训练数据中准确估计时，用 2-gram 来替代，同样，当 2-gram 的值不能从训练语料中准确估计时，可以用 1-gram 的值来代替。插值公式：

$$p(w_3 \mid w_1 w_2) = \lambda_3 p'(w_3 \mid w_1 w_2) + \lambda_2 p'(w_3 \mid w_2) + \lambda_1 p'(w_3)$$

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

- $\lambda_1, \lambda_2, \lambda_3$ **的确定**：

将训练语料分为两部分，即从原始语料中删除一部分作为留存数据(heldout data)**。**

第一部分用于估计 $p'(w_3 \mid w_1 w_2)$ $p'(w_3 \mid w_2)$ **和** $p'(w_3)$ **。**

第二部分用于计算 $\lambda_1, \lambda_2, \lambda_3$**：使语言模型对留存数据的困惑度最小。**

内 容 提 要

4.1 语言模型基本概念

4.2 语言模型参数估计

4.3 语言模型性能评价

4.4 语言模型应用

4.5 改进的语言模型

4.3 语言模型性能评价

目前主要有两种评价方法：

1. **实用方法：**通过查看该模型在实际应用（如拼写检查、机器翻译）中的表现来评价，**优点**是直观、实用，**缺点**是缺乏针对性、不够客观。

4.3 语言模型性能评价

2. 理论方法：

迷惑度/困惑度/混乱度（preplexity），其基本思想是给测试集赋予较高概率值（低困惑度）的语言模型较好，公式如下

平滑的 n-gram 模型句子的概率: $p(s)=\prod_{i=1}^{m+1} p(w_i \mid w_{i-n+1}^{i-1})$

假定测试语料 T 由 l_T 个句子构成（t_1, …, t_{l_T}），则整个测试集的概率为：

$$p(T)=\prod_{i=1}^{l_T} p(t_i)$$

模型 $p(w_i \mid w_{i-n+1}^{i-1})$ 对于测试语料的交叉熵：

$$H_p(T)=-\frac{1}{W_T}\log_2 p(T)$$

其中，W_T 是测试文本 T 的词数。

模型 p 的困惑度 $PP_p(T)$ 定义为：

$$PP_p(T)=2^{H_p(T)}$$

4.3 语言模型性能评价

n-gram 对于英语文本的困惑度范围一般为 50～1000，对应于交叉熵范围为 6 - 10 bits/word。

内 容 提 要

4.4 语言模型应用

语言模型的用途

决定哪一个词序列的可能性更大

已知若干个词，预测下一个词

....

4.4 语言模型应用

应用示例1：

1. 美联储主席本·伯南克昨天告诉媒体 7000 亿美元的救助资金将给上百家银行、保险公司和汽车公司。

2. 本·伯南克美联储主席昨天7000 亿美元的救助资金告诉媒体将借给银行、保险公司和汽车公司上百家。

3. 联主美储席本·伯诉体南将借天的救克告媒昨助资金70元亿00 美给上百百百家银保行、汽车险公司公司和。

4.4 语言模型应用

按 n-gram 模型计算：

句概率≈ 10^{-20}

1. 美联储主席本·伯南克昨天告诉媒体 7000 亿美元的救助资金将给上百家银行、保险公司和汽车公司。

句概率≈ 10^{-25}

2. 本·伯南克美联储主席昨天7000 亿美元的救助资金告诉媒体将借给银行、保险公司和汽车公司上百家。

句概率≈ 10^{-70}

3. 联主美储席本·伯诉体南将借天的救克告媒昨助资金70元亿00 美给上百百百家银保行、汽车险公司公司和。

结论： 第一个句子最有可能

4.4 语言模型应用

应用示例2：　　音字转换问题

给定拼音串：ta shi yan jiu sheng wu de

可能的汉字串：踏实研究生物的
他实验救生物的
他使烟酒生物的
他是研究生物的
… …

4.4 语言模型应用

解：

CString = {踏实研究生物的，他实验救生物的，他是研究生物的，

他使烟酒生雾的, …}

使用 2-gram：

$p(\boldsymbol{CString}_1)$ =p(踏实|<BOS>)×p(研究|踏实)×

p(生物|研究)×p(的|生物)×p(<EOS>|的)

$p(\boldsymbol{CString}_2)$ =p(他|<BOS>)×p(实验|他)×p(救|实验)×

p(生物|救)×p(的|生物)×p(<EOS>|的)

……

选择概率最大的字串

4.4 语言模型应用

应用示例3：　已知若干个词，预测下一个词

基于 *n*-gram 的智能狂拼、微软拼音输入法等

内 容 提 要

4.5 改进的语言模型

n-gram 存在问题：

1. 在训练语言模型时对语料敏感，训练参数难以反映不同领域之间在语言使用规律上的差异。
2. 在自然语言中，常出现某些在文本中通常很少出现，但在某局部文本中大量出现的情况。

需要能根据具体情况**动态调整**语言模型中概率分布参数的**动态、自适应语言模型**。

4.5 改进的语言模型

◆ 基于缓存的语言模型 (Cache-based LM)

问题： 在文本中刚刚出现过的词在后边的句子中再次出现的可能性往往较大，比 标准的 *n*-gram 模型预测的概率要大。

自适应方法：

-- 将K个最近出现过的词存于一个缓存中，作为独立的训练数据。

-- 通过这些数据，计算动态频度分布数据。

-- 将动态频度分布数据与静态分布数据（由大规模语料训练得到）通过线性插值的方法结合。

$$\hat{p}_{Cache}(w_i \mid w_1^{i-1}) = \frac{1}{K} \sum_{j=i-K}^{i-1} I_{\{w_j = w_i\}}$$

$$\hat{p}(w_i \mid w_1^{i-1}) = \lambda\, \hat{p}_{Cache}(w_i \mid w_1^{i-1}) + (1-\lambda)\, \hat{p}_{n-gram}(w_i \mid w_{i-n+1}^{i-1})$$

$0 < \lambda < 1$　　插值系数λ 可以通过EM算法求得。

4.5 改进的语言模型

◆基于混合方法的语言模型

问题：由于大规模训练语料来自不同领域，在主题(topic)、风格(style)都有一定的差异，而测试语料一般是同源的，为了获得最佳性能，语言模型必须适应各种不同类型的语料对其性能的影响。

4.5 改进的语言模型

自适应方法：

-- 将训练语料按来源、主题或类型等聚类(设为*n*类)，语言模型划分成 *n* 个子模型 $M_1, M_2, \ldots, M_n$；

-- 确定适当的训练语料子集，并利用这些语料建立特定的语言模型；

-- 在模型运行时识别测试语料的主题或主题的集合；

-- 整个语言模型的概率通过下面的线性插值公式计算得到：

$$\hat{p}(w_i \mid w_1^{i-1}) = \sum_{j=1}^{n} \lambda_j \hat{p}_{M_j}(w_i \mid w_1^{i-1})$$

其中，$0 \le \lambda_j \le 1$ $\sum_{j=1}^{n} \lambda_j = 1$

λ值可以通过 EM 算法计算出来**(根据困惑度最小原则)**

4.5 改进的语言模型

语言模型变种：

- Class-based N-gram Mode
- Topic-based N-gram Mode
- Cache-based N-gram Model
- Hybrid
- 指数语言模型：
- 神经网络语言模型
- ……

参考文献：

宗成庆，统计自然语言处理（第2版）课件

吴军，数学之美，人民邮电出版社

在 此 表 示 感 谢 ！

信息内容安全技术国家工程实验室
National Engineering Laboratory for Information Security Technologies

谢谢各位！