作业

共计四次作业

NLP入门推荐

- 1.Stanford: **CS224n: Natural Language Processing with Deep Learning** http://web.stanford.edu/class/cs224n/ 课程+作业 最好做一遍
- 2.吴恩达机器学习课程 + 统计学习方法(书)：
  https://www.bilibili.com/video/BV164411b7dx/
- 3.Python + Pytorch (工具)
- 4.几篇NLP经典综述（确定兴趣方向）
- 5.Arxiv 知乎每天逛一逛 （紧跟前沿热点）
- 如果对NLP or IR感兴趣(硕/博)， 欢迎联系宋老师

作业介绍

- 1. 第一次作业：3必做　　DDL：9.23
- 2. 第二次作业：3必做+1选做 DDL：10.13
- 3. 第三次作业：3必做　　DDL：10.20
- 4. 第四次作业：2必做　　DDL：10.27

- 提示：大家注意截止时间，仔细审题，按照要求来，要有自己独立的思考

第一次作业内容

- 1.请说明如下句子有多少种不同的含义?

He drew one card. 英译汉

(1)他画一张卡片

(2)他拉一张纸牌

第一次作业内容

- 1.请说明如下句子有多少种不同的含义?

咬死猎人的狗　？断句

(1)一只狗把猎人咬死了

(2)X X把猎人的狗咬死

第一次作业内容

- 1.请说明如下句子有多少种不同的含义?

这座碑是为纪念反对共产主义者叛乱中牺牲的英雄而建立的。(摘自马来西亚旅游介绍资料)

(1)…反对[共产主义者叛乱]…

(2)… [反对共产主义者]叛乱…

第一次作业内容

- **2.试比较汉英句子中地点状语位置的差异。**

(1)汉语句子中的地点状语一般在谓语动词的前面，而英语句子中的地点状语一般在谓语动词的后面。

如：我在北京理工大学学习。

I am studying in BIT.

第一次作业内容

- **2.试比较汉英句子中地点状语位置的差异。**

(2)地点状语是偏正短语时，汉语中一般范围大的在前(左)面，范围越小、越具体的修饰词离中心词越近，英文中正好相反。

如：他在三层左边那个屋子里的桌子上放了一本书。

He put a book on the desk in the left room of the third floor.

第一次作业内容

- **3.下列语言中哪些为自然语言?**

世界语、C语言、鸟语、甲骨文

自然语言：指人类社会发展过程中自然产生的语言，而不是人为编造的语言,如程序语言等。

第二次作业内容

- **1. 什么是词？谈谈你对“词”这个概念的认识。**

  词是自然语言中能够独立运用的最小单位，是自然语言处理的基本单位。

第二次作业内容

- **2. 写出汉语词语重叠形式的分析规则。**

重叠形式：形如AA、AAB、ABB、AABB、AABC、ABAC、ABCC的词。

汉语词语重叠形式的分析规则为合并原则:语义上无法由组合成分直接相加而得到的字串应该合并为一个分词单位。

（分析+示例说明）

第二次作业内容

- **3.试从互联网上找一篇字数在2000字左右的中文文章，进行人工分词，并列举、归纳碰到的问题。**

分词规范

分词歧义

未登录词识别

体会中文分词的难处，感兴趣同学可以进一步学习相关内容

中文分词算法：基于规则/机器学习/深度学习中文分词

第二次作业内容

- **4.编写程序实现一个有限状态自动机用于识别缩写 {he, she}’s 是 he/she has 还是 he/she is。（选做题，不做不扣分）**

有限状态自动机拥有有限数量的状态，每个状态可以迁移到零个或多个状态，输入字串决定执行哪个状态的迁移。有限状态自动机可以表示为一个有向图。

如果{he/she}’s后面跟的是-ing形式、a、形容词，则说明是he/she is,如：He’s a nurse或者He’s cute.

如果{he/she}’s后面跟的是过去分词，则说明是he/she has,如：He’s been in town for months.

第二次作业内容

有限状态自动机例子

构造一个有限状态自动机M,它能识别{0，1}上的语言$L =$ $\{x000y: x, y \in \{0,1\}\}$

分析:

语言L的特点是语言中的每个串都包含连续的3个0，故FSM的状态及其意义如下:

第二次作业内容

有限状态自动机例子

(1)q_0:有限状态自动机的开始状态，也是重新寻找子串000时的状态；

(2)q_1:有限状态自动机读到第一个0，有可能是子串000的第一个0；

(3)q_2:有限状态自动机在q_1后又读到一个0；

(4)q_3:有限状态自动机在q_2后又读到一个0，这是唯一的接收状态。

第二次作业内容

有限状态自动机例子

因此，状态转移函数为：

$$\delta(q_0, 0) = q_1 \qquad \delta(q_0, 1) = q_0$$

$$\delta(q_1, 0) = q_2 \qquad \delta(q_1, 1) = q_0$$

$$\delta(q_2, 0) = q_3 \qquad \delta(q_2, 1) = q_0$$

$$\delta(q_3, 0) = q_3 \qquad \delta(q_3, 1) = q_3$$

接收状态为$F = \{q_3\}$

第三次作业内容

- 1.试构造一个汉语词性标注的实例，说明用Viterbi算法进行词性标注的过程。

维特比算法

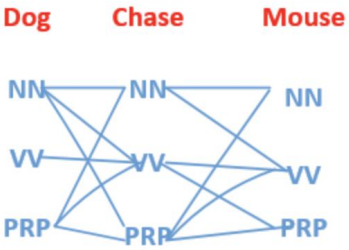
- 维特比算法说白了就是动态规划实现最短路径，只要知道“动态规划可以降低复杂度”这一点就可以轻松理解维特比算法

- 维特比算法之所以重要，是因为凡是使用隐含马尔可夫模型描述的问题都可以用它来解码，包括今天的数字通信、语音识别、机器翻译、拼音转汉字、分词等。——《数学之美》

维特比算法-词性标注

- 我们有一句已经分好词的句子：dog chase mouse. 那么我们就可以进行词性标注为：

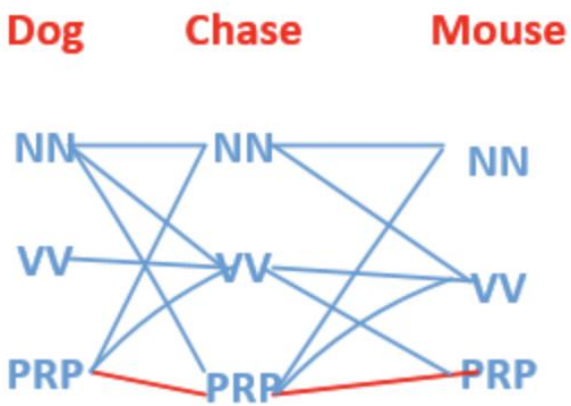
dog chase mouse
nn vv nn



- 其中nn为名词，vv为动词
- 总共有27条路径，那么如何得到我们Dog chase mouse的最优路径呢？

$$P(T_{\text{第一条路径}} \mid Words) = p(Dog \mid NN) * p(NN \mid NN) * p(Chase \mid NN) * p(NN \mid NN) * p(Mouse \mid NN)$$

- 所求的路径对应如下图红色线条所示：



Viterbi算法:提高效率之道

- Viterbi算法[1]是一种动态规划方法(dynamic programming)
- 如果当前节点在最优路径上,那么,不管当前节点的后续路径如何,当前节点的来源路径必定是最优的。
- 最优路径的求解可以迭代进行。

[1] Viterbi, A., 1967, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory Vol.13 No.2, pp260–269

词性转移矩阵(转移概率)给定

	C	f	m	n	p	q	r	v
c	736	700	3971	43250	9253	53	7776	40148
f	900	475	4569	7697	2968	278	1290	26951
m	547	1470	17505	46001	1722	139653	305	13778
n	55177	50571	27918	277181	43023	404	9769	221776
p	47	2664	14131	78251	3363	142	27249	36807
q	732	7845	4506	52310	2451	176	760	13288
r	2055	1225	12820	43953	11229	7681	3572	53391
v	13715	14843	70914	221796	44651	3226	46697	19196

词性频度表 给定

词性	频次
c	168350
f	110878
m	270381
n	1539367
p	269186
q	155374
r	214942
v	1193317

词性标记总频次:
7284443

词语/词性频度表(用于估算输出概率)

词语	词性	频次
把	p	9877
把	q	290
把	n	2
把	v	208
这	r	21990
篇	q	706
报道	v	4040
报道	n	420

词语	词性	频次
编辑	n	243
编辑	v	100
一	m	20672
一	c	2229
下	f	6313
下	q	161
下	v	2271

维特比算法-词性标注

算法(维特比算法)

输入：模型 $\lambda=(A,B,\pi)$ 和观测 $O=(o_1,o_2,\cdots,o_T)$；

输出：最优路径 $I^*=(i_1^*,i_2^*,\cdots,i_T^*)$。

(1)初始化

N代表的是词性的个数

$$\delta_1(i)=\pi_i b_i(o_1),\quad i=1,2,\cdots,N$$

首先进行初始化

$$\psi_1(i)=0,\quad i=1,2,\cdots,N$$

(2)递推。对 $t=2,3,\cdots,T$

当前节点的选取是选取前面到该点转移概率最大的那个

$$\delta_t(i)=\max_{1\le j\le N}[\delta_{t-1}(j)a_{ji}]b_i(o_t),\quad i=1,2,\cdots,N$$

$$\psi_t(i)=\arg\max_{1\le j\le N}[\delta_{t-1}(j)a_{ji}],\quad i=1,2,\cdots,N$$

记录路径的时候记录最大概率的转移上层起始点

(3)终止

挑选最大的那个作为我们的选择

$$P^*=\max_{1\le i\le N}\delta_T(i)$$

$$i_T^*=\arg\max_{1\le i\le N}[\delta_T(i)]$$

(4)最优路径回溯。对 $t=T-1,T-2,\cdots,1$

$$i_t^*=\psi_{t+1}(i_{t+1}^*)$$

回溯回去找最优解

求得最优路径 $I^*=(i_1^*,i_2^*,\cdots,i_T^*)$。

Viterbi算法词性标注示例

- **把/p-q-n-v 这/r 篇/q 报道/v-n 编辑/v-n 一/m-c 下/v-q-f**

把/p-q-n-v →这/r

Delta(这/r)1=a12(把/p→这/r)*b2(这/r) = (27249/269186)*(21990/214942)
=0.01036 √

Delta(这/r)2=a12(把/q→这/r)*b2(这/r) =
(760/155374)*(21990/214942)
=5e-4

Delta(这/r)3=a12(把/n→这/r)*b2(这/r)= (9769/1539367)*(21990/214942)
=6.49e-4

Delta(这/r)4=a12(把/v→这/r)*b2(这/r)= (46697/1193317)*(21990/214942)
=0.004

Viterbi算法词性标注示例

这/r →篇/q

Delta(篇)只有一个,略去

篇/q →报道/v-n

Delta(报道/n)1=a12(篇/q→报道/n)*b2(报道/n)= (52310/155374)*(420/1539367)
=9.1857e-5

Delta(报道/v)1=a12(篇/q→报道/v)*b2(报道/v)= (13288/155374)*(4040/1193317)
=2.8954e-4 √

Viterbi算法词性标注示例

报道/v-n →编辑/v-n

Delta(编辑/n)1=Delta(报道/n)1*a23(报道/n→编辑/n)*b3(编辑/n)
=9.1857e5*(277181/1539367)*(243/1539367)=2.6e-9

Delta(编辑/n)2=Delta(报道/v)1*a23(报道/v→编辑/n)*b3(编辑/n)
=2.8954e-4*(221796/1193317)*(243/1539367) =8.49e-9 √

Delta(编辑/v)1=Delta(报道/n)1*a23(报道/n→编辑/v)*b3(编辑/v)
=9.1857e-5*(221776/1539367)*(100/1193317) =1.1e-9

Delta(编辑/v)2=Delta(报道/v)1*a23(报道/v→编辑/v)*b3(编辑/v)
=2.8954e-4*(191967/1193317)*(100/1193317) =3.9e-9 √

Viterbi算法词性标注示例

编辑/v-n → 一/m-c

Delta(一/m)1=Delta(编辑/n)2*a34(编辑/n→一/m)*b4(一/m)
=8.49e−9*(27918/1539367)*(20672/270381) =1.18e−11

Delta(一/m)2=Delta(编辑/v)2*a34(编辑/v→一/m)*b4(一/m) √
=3.9e−9*(70914/1193317)*(20672/270381) =1.77e−11

Delta(一/c)1=Delta(编辑/n)2*a34(编辑/n→一/c)*b4(一/c)
=8.49e−9*(55177/1539367)*(2229/168350) =4e−12 √

Delta(一/c)2=Delta(编辑/v)2*a34(编辑/v→一/c)*b4(一/c)
=3.9e−9*(13715/1193317)*(2229/168350) =5.9e−13

Viterbi算法词性标注示例

一/m-c → 下/v-q-f

Delta(下/v)1=Delta(一/m)2*a45(一/m→下/v)*b5(下/v)
=1.77e-11*(13778/270381)*(2271/1193317) =1.7e-15

Delta(下/v)2=Delta(一/c)1*a45(一/c→下/v)*b5(下/v)
=4e-12*(40148/168350)*(2271/1193317) =1.8e-15

Delta(下/q)1=Delta(一/m)2*a45(一/m→下/q)*b5(下/q)
=1.77e-11*(139653/270381)*(161/155374) =9.47e-15 √

Delta(下/q)2=Delta(一/c)1*a45(一/c→下/q)*b5(下/q)
=4e-12*(53/168350)*(161/155374) =1.3e-18

Delta(下/f)1=Delta(一/m)2*a45(一/m→下/f)*b5(下/f)
=1.77e-11*(1470/270381)*(6313/110878) =5.47e-15

Delta(下/f)2=Delta(一/c)1*a45(一/c→下/f)*b5(下/f)
=4e-12*(700/168350)*(6313/110878) =9.47e-16

Viterbi算法词性标注示例

一→下

Delta(下/q)1=Delta(一/m)2*a45(一/m→下/q)*b5(下/q)

=1.77e-11*(139653/270381)*(161/155374) =9.47e-15√

最优路径回溯：下/q→一/m

编辑→一

Delta(一/m)2=Delta(编辑/v)2*a34(编辑/v→一/m)*b4(一/m)

=3.9e-9*(70914/1193317)*(20672/270381) =1.77e-11√

最优路径回溯：下/q→一/m→编辑/v

报道→编辑

Delta(编辑/v)2=Delta(报道/v)1*a23(报道/v→编辑/v)*b3(编辑/v)

=2.8954e-4*(191967/1193317)*(100/1193317) =3.9e-9√

最优路径回溯：下/q→一/m→编辑/v→报道/v

Viterbi算法词性标注示例

篇→报道

Delta(报道/v)1=a12(篇/q→报道/v)*b2(报道/v)=
(13288/155374)*(4040/1193317) =2.8954e-4

最优路径回溯：下/q→一/m→编辑/v→报道/v→篇/q

这→篇: Delta(篇)只有一个

最优路径回溯：下/q→一/m→编辑/v→报道/v→篇/q→这/r

把→这

Delta(这/r)1=a12(把/p→这/r)*b2(这/r) =
(27249/269186)*(21990/214942) =0.01036√

最优路径回溯：下/q→一/m→编辑/v→报道/v→篇/q→这/r→把/p

把/p 这/r 篇/q 报道/v 编辑/v 一/m 下/q

第三次作业内容

- **2.对汉语中的兼类词进行分析，撰写词的兼类问题的分析报告，尝试给出词类判定的语言学规则**

兼类词就是一个词具有几种词类语法特征的词(例如:“思考”既有名词又有动词词性)。词性标注时，句子中兼类词的词性根据上下文唯一地确定下来。兼类词分析，需要使用词典来进行统计，统计其中词语兼类的数量。

词类判定的语言学规则，可以根据自己的经验给出。规则的形式可参考课件中山西大学的消歧规则，规则数量不少于5条即可。

第三次作业内容

- **3.了解目前常见的几种汉语词性标注集，比较它们的差异，作说明分析。**

《PFR人民日报标注语料库》

《现代汉语语料库加工规范——词语切分与词性标注》

《计算所ICTCLAS 3.0汉语词性标记集》

《HanLP词性标注集》

《BosonNLP词性标注》

第三次作业内容

- **3.了解目前常见的几种汉语词性标注集，比较它们的差异。**

《PFR人民日报标注语料库》:除26个基本词类标记外,从语料库应用的角度，增加了专有名词（人名nr、地名ns、机构名称nt、其他专有名词nz）；从语言学角度也增加了一些标记，总共使用了40多个标记。

《现代汉语语料库加工规范——词语切分与词性标注》：在上述标注集的基础上添加了区别语素(bg)等

第三次作业内容

- **3.了解目前常见的几种汉语词性标注集，比较它们的差异。**

《计算所ICTCLAS 3.0汉语词性标记集》:由于中科院计算所的汉语词法分析器主要采用北大《人民日报》语料库进行参数训练，因此本词性标记集主要以北大《人民日报》语料库的词性标记集为蓝本，并参考了北大《汉语语法信息词典》中给出的汉语词的语法信息。

《HanLP词性标注集》:HanLP使用的HMM词性标注模型训练自2014年人民日报切分语料，随后增加了少量98年人民日报中独有的词语。

第三次作业内容

- **3.了解目前常见的几种汉语词性标注集，比较它们的差异。**

《BosonNLP词性标注》:BosonNLP词性标注集是基于《北京大学现代汉语语料库基本加工规范》和《计算所汉语词性标记集》修改得到的。与最初《北京大学现代汉语语料库基本加工规范》相比，主要修改有:

1.姓名和起来标”nr”，只有姓单独出现的时候标”nr1”，如“张/nr1 教授/n”　　……

共22个大类，70个标签

第四次作业内容

- **1.阅读参考书中介绍的其它数据平滑方法，自行选择一种进行简要的评价。**

  数据平滑的基本思想可以用一个词语概括：“劫富济贫”

  即使得零概率增值，使得非零概率下调。

加1法(Additive smoothing): 每一种情况出现的次数加1

减值法/折扣法(Discounting):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

①Good-Turing 估计　　②Back-off (后备/后退)方法

③绝对减值法 (Absolute discounting)

④线性减值法 (Linear discounting)

第四次作业内容

- 1.阅读参考书中介绍的其它数据平滑方法，自行选择一种进行简要的评价。

删除插值法 (Deleted interpolation)：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

第四次作业内容

- 1.阅读参考书中介绍的其它数据平滑方法，自行选择一种进行简要的评价。

Kneser-Ney法：一种扩展的绝对减值法，用新的方式建立与高阶分布相结合的低阶分布。前面算法中，通常用平滑后的低阶最大似然分布为低阶分布。然而，只有当高阶分布中具有极少的或没有计数时，低阶分布在组合模型中才是一个重要因素。因此，在这种情况下，应最优化这些参数，以得到较好的性能。

第四次作业内容

- 各种平滑方法的详细介绍和比较请参阅：

Chen, Stanley F. and Joshua Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Model. Available from the website:

http://www-2.cs.cmu.edu/~sfc/html/publications.html

第四次作业内容

- **2.使用讲义中提到的2个语言模型工具（任选一个），对人民日报语料训练语言模型。**

可以使用的语言建模工具：

SRI语言模型工具： http://www.speech.sri.com/projects/srilm/

CMU-Cambridge语言模型工具：

http://www.speech.cs.cmu.edu/SLM/toolkit_documentation.html

可在阅读“自然语言处理入门”一书的“3.2 中文分词语料库”和“3.3 训练”后，使用HanLP提供的接口进行建模。

第四次作业内容

- 统计语言模型:用来计算一个句子的概率的模型

给定一个语句S = $w_1 w_2 \ldots w_N$,它的概率可以表示为:

$$p(s) = p(w_1) \cdot p(w_2 \mid w_1) \cdot p(w_3 \mid w_1 w_2) \cdot \ldots \cdot p(w_m \mid w_1 \ldots w_{m-1})$$

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

第四次作业内容

$$p(s) = p(w_1) \times p(w_2 \mid w_1) \times p(w_3 \mid w_1 w_2) \times \ldots \times p(w_m \mid w_1 \ldots w_{m-1})$$

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

- 可是这样的方法存在两个致命的缺陷：
  - 参数空间过大:条件概率P(wn|w1,w2,..,wn-1)的可能性太多 无法估算，不可能有用
  - 数据稀疏严重：对于非常多词对的组合，在语料库中都没有出现，依据最大似然估计得到的概率将会是0。

第四次作业内容

- 马尔可夫假设：
  - 为了解决参数空间过大的问题。引入了马尔科夫假设：随意一个词出现的概率只与它前面出现的有限的一个或者几个词有关。
    - 如果一个词的出现与它周围的词是独立的，那么我们就称之为unigram也就是一元语言模型：
    - 如果一个词的出现仅依赖于它前面出现的一个词，那么我们就称之为bigram：
    - 假设一个词的出现仅依赖于它前面出现的两个词，那么我们就称之为trigram：

第四次作业内容

- n元模型

基于马尔科夫假设（Markov Assumption）：下一个词的出现仅依赖于它前面的一个或几个词。

1-gram: $p(w_1, w_2, \ldots, w_T) = \prod_{i=1}^{T} p(w_i)$

2-gram: $p(w_1, w_2, \ldots, w_T) = \prod_{i=1}^{T} p(w_i \mid w_{i-1})$

3-gram: $p(w_1, w_2, \ldots, w_T) = \prod_{i=1}^{T} p(w_i \mid w_{i-2}, w_{i-1})$

第四次作业内容

- 最大似然估计：用相对频率计算概率的方法
  - 最大似然估计的思想在于：对于给定的观测数据X，我们希望从所有的参数θ中找出能最大概率生成观测数据的参数θ^*作为估计结果

如果$\{s_1, s_2, \ldots, s_n\}$是一个试验的样本空间，在相同情况下重复试验N次，观察到样本$s_k (1 \le k \le n)$的次数为$n_N(s_k)$,那么，s_k在这N次试验中的相对频率为

$$q_N(s_k) = \frac{n_N(s_k)}{N}$$

第四次作业内容

- 最大似然估计：用相对频率计算概率的方法

由于$\sum_{k=1}^{n} n_N(s_k) = N$,因此$\sum_{k=1}^{n} q_N(s_k) = 1$

当N越来越大时，相对频率$q_N(s_k)$就越来越接近s_k的概率$P(s_k)$。事实上，$\lim_{N\to\infty} q_N(s_k) = P(s_k)$

因此，通常用相对频率作为概率的估计值。

第四次作业内容

- 语言模型的训练

模型的训练也称为模型的参数估计，参数可以用下式估计：

$$p(w_i \mid w_{i-n+1}, \ldots, w_{i-1}) = \frac{c(w_{i-n+1}, \ldots, w_{i-1}, w_i)}{c(w_{i-n+1}, \ldots, w_{i-1})}$$

N元模型以词语为基本单位：分词

在分好词的语料上统计n元对的出现次数

使用最大似然估计的方法对参数进行估计

第四次作业内容

- 困惑度（perplexity）:评价一个语言模型的好坏

基本思想：给测试集的句子赋予较高概率值的语言模型较好,当语言模型训练完之后，测试集中的句子都是正常的句子，那么训练好的模型就是在测试集上的概率越高越好，公式如下：

$$PP(W) = P(w_1, w_2, \ldots w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1, w_2, \ldots w_N)}}$$

由公式可知，**句子概率越大，语言模型越好，困惑度越小**

第四次作业内容

- SRILM 语言模型工具介绍，其主要目标是对语言模型的估计与评测
- SRILM的全称是Stanford Research Institute Language Modeling Toolkit，SRILM是一个应用比较广泛的统计和分析语言模型的工具，它被用来构建和应用统计语言模型，主要用于语音识别，统计标注和切分，以及机器翻译等工作。
- SRILM是生成基于ngram语法的语言模型

SRILM生成的语言模型格式为ARPA格式的ngram语言模型格式

第四次作业内容

- SRILM 语言模型工具教程
- 安装

在github中发现1.7.1版本

```
[root@vmonline ~]# git clone https://github.com/gsayer/SRILM.git
[root@vmonline ~]# cd SRILM/
[root@vmonline SRILM(master)]# mkdir /usr/local/srilm-1.7.1 # 安装目录
[root@vmonline SRILM(master)]# tar -xvf srilm-1.7.1.tar.gz -C /usr/local/srilm-1.7.1/
[root@vmonline SRILM(master)]# cd /usr/local/srilm-1.7.1/
[root@vmonline srilm-1.7.1]# export SRILM=$(pwd) # 指定SRILM源码目录
[root@vmonline srilm-1.7.1]# make World # 编译完成即可
```

第四次作业内容

安装

版本编译完成后，配置环境变量，指定SRILM的bin目录

```
[root@vmonline srilm-1.7.1]# vi /etc/profile # 在文件中合适位置设置以下环境变量
SRILM=/usr/local/srilm-1.7.1
export PATH=$SRILM/bin/i686-m64:$PATH
[root@vmonline srilm-1.7.1]# source /etc/profile
```

第四次作业内容

训练

```
[root@vmonline ~]# ngram-count -help # 查看参数使用，其中包含了各种平滑算法
[root@vmonline novel]# ngram-count  -text train.txt  -order 3 -lm train.bin -interpolate -unk -write-binary-lm
```

```
记录ngram-count命令的几个参数解析：
  -order arg  最大的模型阶数，默认值：3
  -read arg 读取count文件
  -text arg 读取分词后的文本文件
  -vocab arg 词汇文件（一行一个词汇），限制text和count文件的单词，没有出现在词典的单词替换为<unk>；如果没有指定该参数，训练文本中所有的词汇将会被自动加入词典
  -limit-vocab 只限制count文件的单词（对text文件无效），没有出现在词典里面的count将会被丢弃
  -write-vocab arg 输出词典到指定文件中
  -lm arg 输出语言模型
  -unk 对于不在词汇文件中的词使用<unk>代替
  -write-binary-lm 输出二进制的语言模型
  -sort 输出语言模型gram排序
```

第四次作业内容

- ARPA格式

三元语言模型的综合格式：

```
\data
ngram 1=nr # 1-gram总数量
ngram 2=nr # 2-gram总数量
ngram 3=nr # 3-gram总数量

\1-grams:
pro_1 word1 back_pro1

\2-grams:
pro_2 word1 word2 back_pro2

\3-grams:
pro_3 word1 word2 word3

\end\
```

第一项表示ngram的条件概率，就是 $P(wod_n|wod_1, wod_2, \ldots, wod_{n-1})$；

第二项表示ngram的词 wod_1^j；

最后一项是回退的权重。

第四次作业内容

- 如何使用该模型来求词语的条件概率？

例如求一个3元的条件概率$pro_3(word\ 3|word\ 1, word\ 2)$：

```
if(存在(word1,word2,word3)的三元模型){
    return pro_3(word1,word2,word3) ;  // 模型中word1,word2,word3对应的3grams概率
}else if(存在(word1,word2)二元模型){  // 不存在则回退
    return back_pro2(word1,word2)*pro_2(word3|word2) ;  // 模型中word1,word2的回退权重 * 模型中word2,word3的2grams概率
}else{  // 依然不存在则再回退，这样误差就有点大了
    return pro_2(word3|word2);  // 模型中word2,word3的2grams概率
}
```

第四次作业内容

- 从上面可知，如果模型中没有word1,word2,word3对应的3grams概率，则都需要找模型中的2grams概率，也就是求一个2元的条件概率$po_2(wod\ 2|wod\ 1)$：

```
if(存在(word1,word2)的二元模型){
    return pro_2(word1,word2);  //模型中word1,word2对应的2grams概率
}else{  // 回退
    return back_pro2(word1)*pro_1(word2) ;  // 模型中word1的回退权重 * 模型中word2的1grams概率
}
```

第四次作业内容

实践

1.训练语料，人民日报，如下：

```
[root@vmlinux ngrams(master)]# less RenMinData.txt_utf8
１９８６年 ，
十亿 中华 儿女 踏上 新 的 征 程 。
过去 的 一 年 ，
是 全国 各族 人民 在 中国 共产党 领导 下 ，
在 建设 有 中国 特色 的 社会主义 道路 上 ，
坚持 改革 、 开放 ，
团结 奋斗 、 胜利 前进 的 一 年 。
城乡 经济 体制 改革 向 纵深 稳步 发展 ，
.......
```

第四次作业内容

实践

2.1.训练1，词频统计

```
[root@vmlinux ngrams(master)]# ngram-count -text RenMinData.txt_utf8 -order 3 -write train.count
[root@vmlinux ngrams(master)]# less train.count
来到    343
来到 陕甘       2
来到 陕甘 游击队        2
来到 郑州       3
来到 郑州 机动  1
来到 郑州 铁路局        1
来到 郑州 市    1
来到 前线       1
来到 前线，     1
来到 法国       2
来到 法国 旅游  1
.....
```

第四次作业内容

- 实践

2.2.训练2，-kndiscount为修正Kneser-Ney打折法

```
[root@vmlinux ngrams(master)]# ngram-count -read train.count -order 3 -lm train.lm  -kndiscount -unk
[root@vmlinux ngrams(master)]# ll train.lm -h
-rw-r--r-- 1 root root 28M May 13 17:12 train.lm
```

第四次作业内容

实践

3.在Python3使用arpa模块加载模型

```
In [1]: import arpa
In [2]: time models=arpa.loadf('train.lm')          # 1.0GB->578MB, 28M的文件占用这么多内存，而且加载时间也长
CPU times: user 5.73 s, sys: 245 ms, total: 5.97 s
Wall time: 5.97 s
In [4]: models
Out[4]: [<arpa.models.simple.ARPAModelSimple at 0x7f1d62f9c910>]
In [5]: lm=models[0]
In [6]: lm.vocabulary()  # 词汇表
.....
In [7]: lm.counts()     # 各个ngrams统计数据
Out[7]: [(1, 25008), (2, 851329), (3, 295546)]
In [8]: lm.order()     # 最高阶数
Out[8]: 3
```

第四次作业内容

实践

4.判断词语的条件概率，注意使用空格分隔（或者使用tuple/list），最好3个词语（3gram）

```
In [14]: lm.p('大家 好')  # 小数值，P(好|大家)，2gram
Out[14]: 0.0005890633460723377
In [15]: lm.p('你们 好')
Out[15]: 0.00043275082052490147
In [16]: lm.log_p('你们 好')    # 对数值
Out[16]: -3.3637620999999998
In [19]: lm.p('你们 好 吗')    # P(吗|你们,好)，3gram
Out[19]: 0.002860836249023289
```

第四次作业内容

实践

5.判断句子的概率，注意使用空格分隔（或者使用tuple/list）！

```
In [22]: lm.s('在 改革 、 开放 的 新 时期 ')    # 句子出现的概率，小数值
Out[22]: 3.601964597749519e-15
In [23]: lm.log_s('在 改革 、 开放 的 新 时期 ')   # 对数值，还是对数值直观一点
Out[23]: -14.443460559999991
In [24]: lm.log_s('都 是 为了 完善 社会主义 制度')  # 对数值，还是对数值直观一点
Out[24]: -17.1574447
In [25]: lm.log_s('制度 社会主义 完善 为了 是 都')  # 句子不通顺时出现的概率还是明显小的
Out[25]: -30.013733000000002
In [30]: lm.log_s('只是 在 牧师 协会 眼中 ， 这是 神灵 赐予 他们 的 礼物 。')  # 集外句子
Out[30]: -35.06874904999998
```

第四次作业内容

实践

6.写下arpa如何处理不在词汇表中的词语

比如‘牧师’没有在词汇表中，arpa模块将使用<unk>替代‘牧师’，零概率事件也不会发生。

```
In [30]: lm.log_s('只是 在 牧师 协会 眼中 ， 这是 神灵 赐予 他们 的 礼物 。')
Out[30]: -35.06874904999998
In [45]: lm.log_s('只是 在 <unk> 协会 眼中 ， 这是 神灵 赐予 他们 的 礼物 。')  # 手动替换，概率是一样的
Out[45]: -35.06874904999998
```

第四次作业内容

评估

1.测试语料，直接取人民日报最后1000行数据

```
[root@vmlinux ngrams(master)]# tail RenMinData.txt_utf8 -n 1000 >> test.txt
```

2.评估困惑度

```
[root@vmlinux ngrams(master)]# ngram -lm train.lm -ppl test.txt -unk
file test.txt: 1000 sentences, 10122 words, 0 OOVs  # 0个集外词，正常，因为偷懒训练语料包含了测试语料
0 zeroprobs, logprob= -20640.4 ppl= 71.7489 ppl1= 109.436
```

第四次作业内容

- CMU-Cambridge语言模型工具教程参考：

https://blog.csdn.net/u012637501/article/details/40894947

QA

祝大家国庆长假过的愉快!