2017211382 陈璐

计算语言学

第四次作业

使用经过人工分词后的北京大学《人民日报》标注语料库（含train/valid/test三个文件夹）完成基于马尔科夫模型及smoothing策略的句子概率计算。

对上述语料库做适当处理，如：在每一句开头增加一个句子起始符<BOS>, 在其结尾处增加一个句子结束符<EOS>。（可以认为这两个特殊字符是“虚拟”地加上去的）基于词的**bigram**（一阶马尔科夫模型）计算如下句子的概率（参考lesson4part2课件）：

- 扶贫 开发 工作 取得 很 大 成绩 （句子1）
- 扶贫 开发 工作 得到 很 大 成绩 （句子2）
- 张家口 震区 处处 见 新 联 （句子3）
- 海南 黎族 乡亲 迁 新居 （句子4）

✻ 先给出关于原始数据的一些统计信息：

	Sentence	Token	Bigram
train	40806	907150	912896
valid	5344	117111	117941
test	4459	96460	97180

要求用如下几种smoothing方法：

1）根据在train + valid + test整个语料库中的统计结果，使用adding-的smoothing策略计算上述句子的概率。要求分别测试=0.1及1.0共2组参数。

$$P(w_2 \mid w_1) = \frac{C(w_1, w_2) + \lambda}{C(w_1) + V\lambda}$$

上次作业中不加平滑的句子1与句子2的概率如下：

注：报告中所有的概率输出皆进行了log变换操作

简单回顾一下上次得出的未做平滑操作的句子1和句子2的unigram和bigram的概率，如下，表1-表4：

	句子P	p(扶贫)	p(开发)	p(工作)	P(取得)	p(很)	p(大)	p(成绩)
句子1 不平滑	-22.281685	-3.489591	-3.364652	-2.667120	-3.284575	-3.089503	-2.698475	-3.687770

表1 不做平滑操作的句子1的unigram概率
和句子各部分的unigram概率

	句子P	p(扶贫)	p(开发)	p(工作)	P(得到)	p(很)	p(大)	p(成绩)
句子2 不平滑	-22.474899	-3.489591	-3.364652	-2.667120	-3.477789	-3.089503	-2.698475	-3.687770

表2 不做平滑操作的句子2的unigram概率
和句子各部分的unigram概率

	句子P	p(扶贫|<BOS>)	p(开发|扶贫)	p(工作|开发)	p(取得|工作)	p(很|得到)	p(大|很)	p(成绩|大)	p(<EOS>|成绩)
句子1 不平滑	-14.41511	-3.571557	-0.984466	-2.198657	-1.904212	-2.452553	-0.727501	-2.187319	-0.672867

表3 不做平滑操作的句子1的bigram概率
和句子各部分条件概率

	句子P	p(扶贫|<BOS>)	p(开发|扶贫)	p(工作|开发)	p(得到|工作)	p(很|得到)	p(大|很)	p(成绩|大)	p(<EOS>|成绩)
句子2 不平滑	-15.08798	-3.571557	-0.984466	-2.198657	-2.667640	-2.077973	-0.727501	-2.187319	-0.672867

表4 不做平滑操作的句子2的bigram概率
和句子各部分条件概率

根据上面add-lambda的公式计算各条件概率，并且将各条件概率相乘得到add-lambda smoothing 后的bigram句子概率。其中λ作为参数，设置λ=0.1和λ=1两组参数，得出表5-表8:

	句子P	p(扶贫|<BOS>)	p(开发|扶贫)	p(工作|开发)	p(取得|工作)	p(很|取得)	p(大|很)	p(成绩|大)	p(<EOS>|成绩)
句子1 λ=0.1	-21.42921	-3.632079	-2.201780	-3.288603	-2.436643	-3.464752	-1.584201	-2.742045	-2.079114
句子1 λ=1	-27.90995	-3.879260	-3.166648	-4.145313	-3.284979	-4.271012	-2.522799	-3.584746	-3.055199

表5 做add-λ平滑操作的句子1的bigram概率
和句子各部分条件概率

	句子P	p(扶贫|<BOS>)	p(开发|扶贫)	p(工作|开发)	p(得到|工作)	p(很|得到)	p(大|很)	p(成绩|大)	p(<EOS>|成绩)
句子2 λ=0.1	-22.00131	-3.632079	-2.201780	-3.288603	-3.192966	-3.280527	-1.584201	-2.742045	-2.079114
句子2 λ=1	-28.48236	-3.879260	-3.166648	-4.145313	-3.983949	-4.144450	-2.522799	-3.584746	-3.055199

表6 做add-λ平滑操作的句子2的bigram概率
和句子各部分条件概率

	句子P	p(张家口|<BOS>)	p(震区|张家口)	p(处处|震区)	p(见|处处)	p(新|见)	p(联|新)	p(<EOS>|联)
句子3 λ=0.1	-26.284700	-3.898092	-3.708815	-3.704236	-3.703768	-3.715859	-3.849927	-3.704002
句子3 λ=1	-30.797534	-4.122298	-4.443232	-4.442770	-4.442723	-4.443951	-4.459815	-4.442746

表7 做add-λ平滑操作的句子3的bigram概率
和句子各部分条件概率

	句子P	p(海南|<BOS>)	p(黎族|海南)	p(乡亲|黎族)	p(迁|乡亲)	p(新居|迁)	p(<EOS>|新居)
句子4 λ=0.1	-20.672980	-3.745029	-3.707346	-3.253174	-3.707346	-3.423332	-2.836754
句子4 λ=1	-25.067980	-3.983995	-4.443083	-4.141630	-4.443083	-4.266671	-3.789518

表8 做add-λ平滑操作的句子4的bigram概率
和句子各部分条件概率

结论：

1. 所有加了λ平滑后句子的bigram概率都大幅度下降，特别是做了add one smoothing后句子bigram的概率约等于0（-25～-30的量级）

2. λ大（如λ=1）得到的句子概率和各部分乘子都比λ小（λ=0.1）的概率更小。这是因为，λ大，分配给未见过的bigram概率配比更大，分配给已见过的bigram概率配比小）。实际上，根据计算得到，λ=1和λ=0.1时分配给所有未见过的bigram的概率有99.9%,也就是有99.9%的概率都转移到了未见过的bigram上。

3. 句子1 和 句子2 做了λ（0.1和1两种情况）平滑后得到的句子bigram概率比不做平滑时句子的unigram概率更小，验证了poor estimation is worse than none的说法。

4. add-lambda优点：简单，便于实现；缺点：转移太多概率

2）以train为基础，将valid以及 valid + test分别当做Held out语料库，使用Held out estimation, 计算上述句子的概率。

为了观察在旧文本出现r次的bigram，在新文本（valid / valid + test）中出现的概率如何，使用held out estimator的方法实现。

对于每个bigram w_1w_2:

$C_1(w_1w_2)$ = frequency of w_1w_2 in training data

$C_2(w_1w_2)$ = frequency of w_1w_2 in held out data

$$T_r = \sum_{\{w1w2:C_1(w_1,w_2)=r\}} C_2(w1w2)$$

$$P_{ho}(w_1w_2) = \frac{T_r}{NrN} \quad where \quad C(w_1w_2) = r$$

设Set1: held out为valid，Set2: held out 为 valid + test

	句子P	p(扶贫|<BOS>)	p(开发|扶贫)	p(工作|开发)	p(取得|工作)	p(很|取得)	p(大|很)	p(成绩|大)	p(<EOS>|成绩)
句子1 Set1	-25.23143	-4.956589	-1.992165	-4.682996	-2.743389	-4.762550	-0.912209	-3.617242	-1.564292
句子1 Set2	-22.79108	-4.651546	-1.687122	-4.377953	-2.438346	-4.457507	-0.607166	-3.312198	-1.259249

表1 句子1在held out data的bigram概率
和句子各部分条件概率

	句子P	p(扶贫|<BOS>)	p(开发|扶贫)	p(工作|开发)	p(得到|工作)	p(很|得到)	p(大|很)	p(成绩|大)	p(<EOS>|成绩)
句子2 Set1	-26.39970	-4.956589	-1.992165	-4.682996	-4.506068	-4.168140	-0.912209	-3.617242	-1.564292
句子2 Set2	-23.95935	-4.651546	-1.687122	-4.377953	-4.201024	-3.863096	-0.607166	-3.312198	-1.259249

表2 句子2在held out data的bigram概率
和句子各部分条件

	句子P	p(张家口|<BOS>)	p(震区|张家口)	p(处处|震区)	p(见|处处)	p(新|见)	p(联|新)	p(<EOS>|联)
句子3 Set1	-57.691431	-5.797061	-8.687072	-8.154433	-7.958138	-8.948731	-10.078712	-8.067283
句子3 Set2	-55.556128	-5.492018	-8.382028	-7.849390	-7.653095	-8.643688	-9.773669	-7.762240

表3 句子3在held out data的bigram概率
和句子各部分条件

	句子P	p(海南|<BOS>)	p(黎族|海南)	p(乡亲|黎族)	p(迁|乡亲)	p(新居|迁)	p(<EOS>|新居)
句子4 Set1	-35.999197	-5.480298	-8.510980	-7.414070	-8.343489	-4.201429	-2.048930
句子4 Set2	-34.168938	-5.175255	-8.205937	-7.109027	-8.038446	-3.896386	-1.743887

表4 句子4在held out data的bigram概率
和句子各部分条件

结论：

1. 对于4个句子来说，很明显能看出在held out出现的概率变小了。这就说明这些句子在held out出现的可能性更小，不过这跟held out集的大小也有关系。

2. 对于同一个句子，但是held out集合大小不同，如果held out data范围更大，所得的在held out的概率会更大，从上面4个句子得出来在不同held out 集合上的概率结果可以看出来。

3. 在held out中的$P_{empirical}$比1）中平滑过的句子概率小一些，按书上table 6.4表格中（如下图）$f_{empirical}$是在held out出现的次数，比在training set出现的次数小一些，而在本节表中，从词语条件概率和1）表中的条件概率比较，held out 的概率比training set上的小一个量级，初步考虑的原因是，training set 和 valid set test set的数

据量相差比较大，training set 是 valid set 和 test set 大小的十倍左右。

$r = f_{MLE}$	$f_{empirical}$
0	0.000027
	0.448
2	1.25
3	2.24
4	3.23
5	4.21
6	5.23
-	6.21
8	7.21
9	8.26

4. 但是，对于句子3和句子4，他们在training set没有出现，但是在held out集也有出现的概率，虽然说出现的概率非常小。在计算held out概率时没有做平滑，所以句子3（-57）和句子4（-35）计算出来的概率远远小于在1）中做了add λ平滑的概率。

3）分别在train + valid以及 train + valid + test上，使用Good-Turing estimation，计算上述句子的概率。

$$r^* = (r+1)\frac{N_{r+1}}{N_r}$$

$$if C(w_1w_2) = r > 0 \qquad P_{GT}(w_1w_2) = \frac{r^*}{N} \qquad f_{GT} = r^*$$

$$if C(w_1w_2) = 0 \qquad P_{GT}(w_1w_2) \approx \frac{N_1}{N_0N} \qquad f_{GT} \approx \frac{N_1}{N_0} \quad Where \quad N_0 = V^2 - N$$

$$P(w_2|w_1) = \frac{f_{GT}(w_1w_2)}{C(w_1)}$$

设Set1： train + valid， Set2: train + valid + test

	句子P	p(扶贫|<BOS>)	p(开发|扶贫)	p(工作|开发)	p(取得|工作)	p(很|取得)	p(大|很)	p(成绩|大)	p(<EOS>|成绩)
句子1 Set1	-15.17303	-3.619600	-1.092492	-2.336582	-1.995319	-2.665803	-0.732292	-2.214458	-0.516485
句子1 Set2	-15.60989	-3.598921	-1.030202	-2.366509	-1.893022	-2.690711	-1.130048	-2.228947	-0.671532

表1 Good Turing smoothing 后句子1的bigram概率
和句子各部分条件

	句子P	p(扶贫|<BOS>)	p(开发|扶贫)	p(工作|开发)	p(得到|工作)	p(很|得到)	p(大|很)	p(成绩|大)	p(<EOS>|成绩)
句子2 Set1	-15.49998	-3.619600	-1.092492	-2.336582	-2.765919	-2.222154	-0.732292	-2.214458	-0.516485
句子2 Set2	-16.06276	-3.598921	-1.030202	-2.366509	-2.783234	-2.253373	-1.130048	-2.228947	-0.671532

表2 Good Turing smoothing 后句子2的bigram概率
和句子各部分条件

	句子P	p(张家口|<BOS>)	p(震区|张家口)	p(处处|震区)	p(见|处处)	p(新|见)	p(联|新)	p(<EOS>|联)
句子3 Set1	-18.280420	-3.881672	-2.406425	-1.897119	-1.760899	-2.688971	-3.801460	-1.843873
句子3 Set2	-18.404171	-3.912680	-2.410277	-1.895832	-1.781889	-2.728832	-3.832075	-1.842586

表3 Good Turing smoothing 后句子3的bigram概率
和句子各部分条件

	句子P	p(海南|<BOS>)	p(黎族|海南)	p(乡亲|黎族)	p(迁|乡亲)	p(新居|迁)	p(<EOS>|新居)
句子4 Set1	-21.946403	-3.855800	-5.719547	-4.550755	-5.528478	-1.843873	-0.447950
句子4 Set2	-10.941588	-3.753247	-2.300403	-0.760845	-2.300403	-1.323728	-0.502963

表4 Good Turing smoothing 后句子4的bigram概率
和句子各部分条件

结论：

1. 对于做了Good Turing Smoothing后的句子1和句子2概率可以说是非常接近于未做平滑的句子概率，概率基本处于一个数量级，所以用Good Turing Smoothing的方法可以说是效果非常好了。

2. 在保证出现过的句子的概率接近MLE的同时，对于句子3、句子4这种有些单词没有在training set中出现过的句子，如果用MLE的方法求句子概率，由于有未出

现词，会得到句子概率为0。但是经过了Good Turing Smoothing后，得到的句子概率也还不错，不会因为有些词没有出现过，而使整个句子概率乘积为0.

3. 为了探究在不同大小的训练集的效果，设置了两个set（train+valid / train+valid+test），句子3在两个set的句子概率基本相同，句子4在两个set的差别比较大，在set2上的句子概率远远大于set1上的句子概率。这也是必然的，因为句子4的选取是从test集上直接从原文本中选取的，而valid 和 train 中都没有出现过相似的bigram，所以句子的概率会比set2（包含了test）小很多。