作业说明

任务与提交要求

根据课堂所学内容，参考所给材料，补全n元语言模型算法及采用Good-Turing折扣的Katz回退算法，并在所给数据集上测试。

代码中已给出预处理文本与测试模型的相关代码，仅要求补全模型代码中`TODO`标注的地方。需要填充的位置共7处，预计共需要约30～60行代码。具体包括：

1. 统计各词组（gram）在训练语料中的频数
2. 计算同频词组个数N_r
3. 计算$d(W_{k-n+1}^{k})$
4. 计算$\alpha(W_{k-n+1}^{k-1})$
5. 根据公式计算回退概率
6. 计算概率对数
7. 计算困惑度（PPL）

提交时建议直接提交带有已补全的代码及运行结果的Jupyter Notebook文件；也可以提交单独的已补全的python源代码与报告，报告中要求对代码作必要的说明并展示运行结果。

文件说明

- `ngram.ipynb` - 作业笔记文件
- `ngram_h.py` - 提供的作业源码，内容同笔记文件中的代码相同
- `data` - 指定的训练与测试语料
- `作业说明.pdf` - 本说明文档，介绍本次作业的具体要求，补充相关知识点与推导过程。
- `ngram-discount.pdf` - 作为参考资料的SRILM的`ngram-discount(7)`手册页

补充语法介绍

提供的代码中采用了Python 3.5引入的类型注解语法，此处简要解释：

定义变量或声明函数参数时，可以采用

```
variable: Type
```

这种方式标注变量类型。定义函数时，可以采用

```python
def foo(a: A) -> B:
```

的方式声明参数与返回值类型。该类型注解不是强制的，在运行时不起作用。更详细的类型语法请自行查询Python文档与教程，此处不再介绍。

补充算法介绍

该代码实现了n元语言模型中采用Good-Turing折扣的Katz回退平滑算法。

在课程中已经学习了Katz回退算法的公式：

$$P_{\text{bo}}(w_k|W_{k-n+1}^{k-1}) = \begin{cases} d(W_{k-n+1}^{k})\dfrac{C(W_{k-n+1}^{k})}{C(W_{k-n+1}^{k-1})} & C(W_{k-n+1}^{k}) > 0 \\ \alpha(W_{k-n+1}^{k-1})P_{\text{bo}}(w_k|W_{k-n+2}^{k-1}) & \text{否则} \end{cases}$$

该公式中具体参数如何选择，本说明会在下面简要推导、介绍。

选择α

选取的$\alpha(W_{k-n+1}^{k-1})$应满足

$$\sum_{w_k \in V} P_{\text{bo}}(w_k|W_{k-n+1}^{k-1}) = 1$$

对已知的W_{k-n+1}^{k-1}，记

$$V_+ = \{w_k|C(W_{k-n+1}^{k-1}w_k) > 0\}$$

即对应词组在训练语料中出现过的词的集合；类似的，记

$$V_- = \{w_k|C(W_{k-n+1}^{k-1}w_k) = 0\} = V \setminus V_+$$

即对应词组在训练语料中未出现的词的集合。

则

$$\sum_{w_k \in V_+} P_{\text{bo}}(w_k|W_{k-n+1}^{k-1}) + \sum_{w_k \in V_-} \alpha(W_{k-n+1}^{k-1})P_{\text{bo}}(w_k|W_{k-n+2}^{k-1}) = 1$$

可解得

$$
\begin{aligned}
\alpha(W_{k-n+1}^{k-1}) &= \frac{1-\sum_{w_k\in V_+} P_{\text{bo}}(w_k|W_{k-n+1}^{k-1})}{\sum_{w_k\in V_-} P_{\text{bo}}(w_k|W_{k-n+2}^{k-1})} \\
&= \frac{1-\sum_{w_k\in V_+} P_{\text{bo}}(w_k|W_{k-n+1}^{k-1})}{1-\sum_{w_k\in V_+} P_{\text{bo}}(w_k|W_{k-n+2}^{k-1})}
\end{aligned}
$$

选择d

在课程中已经学习了，Good-Turing折扣（以下简称“折扣”）将出现次数多的词组的概率摊给出现次数少的，会导致出现次数最多的词组的概率变成零，这会带来很大的误差，因此实践中该策略只在低频词上采用。参考SRILM的`ngram-discount(7)`手册页，可以选取$\theta=7$作为是否采用折扣策略的阈值。因此，对$w_k\in V_+$，可以写出如下公式：

$$
d(W_{k-n+1}^{k-1}w_k)=\begin{cases}1 & C(W_{k-n+1}^{k-1}w_k)>\theta \\ d'(W_{k-n+1}^{k-1}w_k) & \text{否则}\end{cases}
$$

此时，由于高频词组没有将概率匀出来给低频词组，因此直接应用原折扣策略，会导致加起来的概率和超过1。这可以应用一种插值策略来解决。

注意到，折扣策略实质上就是将出现次数为$(r+1)$次的词组的和概率P_{r+1}摊给出现次数为r次的词组，因此可以将问题抽象一下：

折扣前，所有出现频率为r的词组的和概率为P_r，不插值直接折扣后变为P_{r+1}。

应用折扣策略前后，高频词组的概率和没有变化，因此零频词组（V_-）与低频词组的概率和也不应因折扣而改变。注意到折扣策略会将P_1的概率和分给零频词组，因此折扣后，低频词组的概率和应为$\sum_{i=2}^{\theta}P_i$，即，插值策略应满足

$$
\sum_{r=1}^{\theta}(\lambda P_{r+1}+(1-\lambda)P_r)=\sum_{r=2}^{\theta}P_i
$$

易解得$\lambda=\frac{P_1}{P_1-P_{\theta+1}}$。

由于

$$
P_r=\sum_{C(W_{k-n+1}^{k-1}w_k)=r}P(w_k|W_{k-n+1}^{k-1})=N_r\frac{r}{C(W_{k-n+1}^{k-1})}
$$

代入可得

$$
\lambda=\frac{N_1}{N_1-(\theta+1)N_{\theta+1}}
$$

因此

$$d'(W_{k-n+1}^{k-1}w_k) = \lambda\frac{(r+1)N_{r+1}}{rN_r} + (1-\lambda)$$

其中$r = C(W_{k-n+1}^{k-1}w_k)$。