NLP Lab01: n-gram Language Model with Good-Turing Discounting and Katz Back-off

n-gram折扣回退算法原理

用w_k表示句子中第k个字，记$W_i^j=\overline{w_iw_{i+1}\dots w_j}$，即当前句子中以$w_j$结尾的$(j-i+1)$-gram。

则在n-gram的前$(n-1)$个字为W_{k-n+1}^{k-1}时，词尾为w_k的条件概率为

$$P_{\mathrm{bo}}\left(w_k\middle|W_{k-n+1}^{k-1}\right)=\begin{cases}d\left(W_{k-n+1}^{k}\right)\dfrac{C\left(W_{k-n+1}^{k}\right)}{C\left(W_{k-n+1}^{k-1}\right)}, & C\left(W_{k-n+1}^{k}\right)>0\\ \alpha\left(W_{k-n+1}^{k-1}\right)P_{\mathrm{bo}}\left(w_k\middle|W_{k-n+2}^{k-1}\right), & \mathrm{otherwise}\end{cases}$$

其中，$d(W_{k-n+1}^{k})$为折扣系数，$\alpha\left(W_{k-n+1}^{k-1}\right)$为回退系数。

- 当训练集中有W_{k-n+1}^{k}这个n-gram时，即$w_k\in V_+=\left\{w_k\middle|C\left(W_{k-n+1}^{k}\right)>0\right\}$时，我们进行discounting处理。
- 当训练集中没有W_{k-n+1}^{k}这个n-gram时，即$w_k\in V_-=V\setminus V_+$时，我们进行back-off处理。

Discounting

我们采用Good-Turing Discounting，即

$$d\left(W_{k-n+1}^{k}\right)=\begin{cases}1, & C\left(W_{k-n+1}^{k}\right)>\theta\\ d'\left(W_{k-n+1}^{k}\right), & 0<C\left(W_{k-n+1}^{k}\right)\le\theta\end{cases}$$

其中折扣系数的相关计算公式如下：

$$\lambda=\frac{N_1^{[W_{k-n+1}^{k-1}]}}{N_1^{[W_{k-n+1}^{k-1}]}-(\theta+1)N_{\theta+1}^{[W_{k-n+1}^{k-1}]}}$$

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

其中 $r = C\left(W_{k-n+1}^{k}\right)$，$N_q^{[W_{k-n+1}^{k-1}]}$表示 出现次数为$q$次 且 开头$(n-1)$个字为$W_{k-n+1}^{k-1}$ 的n-gram的个数，θ为判断是否需要采用折扣策略的阈值。

根据SRILM (http://www.speech.sri.com/projects/srilm/)中的 `ngram-discount(7)` 可知一般选择使用7。

上式的推导如下：

记所有出现频率为r的以W_{k-n+1}^{k-1}开始的词组的和概率为P_r。则

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

$$\text{where } P_r=\sum_{w_k \text{ s.t. } C\left(W_{k-n+1}^{k-1} w_k\right)=r} P\left(w_k \middle| W_{k-n+1}^{k-1}\right)=N_r^{[W_{k-n+1}^{k-1}]} \frac{r}{C\left(W_{k-n+1}^{k-1}\right)}$$

由此可推得上述计算公式。

Back-off

我们采用Katz回退算法，即对$w_k \in V_-$的情况，使用$\alpha\left(W_{k-n+1}^{k-1}\right) P_{\mathrm{bo}}\left(w_k \middle| W_{k-n+2}^{k-1}\right)$来近似$P_{\mathrm{bo}}\left(w_k \middle| W_{k-n+1}^{k-1}\right)$。

其中回退系数$\alpha\left(W_{k-n+1}^{k-1}\right)$的计算公式如下：

$$\alpha\left(W_{k-n+1}^{k-1}\right)=\frac{1-\sum_{w \in V_+} P_{\mathrm{bo}}\left(w_k \middle| W_{k-n+1}^{k-1}\right)}{1-\sum_{w \in V_+} P_{\mathrm{bo}}\left(w_k \middle| W_{k-n+2}^{k-1}\right)}$$

上式的具体推导如下：

选择的$\alpha\left(W_{k-n+1}^{k-1}\right)$应满足

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

$$\text{i.e. } \sum_{w_k \in V_+} P_{\mathrm{bo}}\left(w_k \middle| W_{k-n+1}^{k-1}\right)+\sum_{w_k \in V_-} \alpha\left(W_{k-n+1}^{k-1}\right) P_{\mathrm{bo}}\left(w_k \middle| W_{k-n+2}^{k-1}\right)=1$$

由此可推导上述计算公式。

n-gram语言模型的实现

具体详见 `ngram.ipynb` 或 `ngram_h.py` 。程序中加入的注释解释了各个变量的含义。

最终得到的困惑度（PPL）结果如下：

8693.780642644684
331.10972998040086
4921.741528111271
1607.896405353078
1219.4584822690754
141.72320520636305
4397022.162024288
515.0659925165494
2373.187201414056
7609.707886973888
145.95131196843286
609.0896766368473
9810.215089654144
5076.607436082766
18965.348034596627
13542.03248372126
376.2696786406674
23756.06813777847
7642.519944372413
82.46336642614023
1510.080716626666
769.4242467625158
33119.594458685846
116237.02958117228
14926.892379689782
14310.48191958656
3320.4047934646383
4113.60310503891
1079.0929109195215
407.62766869783763
2193.270993370737
8891.929058381042
5609.474510696962
3419.3965661988905
463.66384566246103
775258.2735668279
12959.445395207813
881.3357966253474
6339.019858902836
1059.106005914286
4502.3878699325305
4265.678462648636
2655.900205257135
129527.0195328782
2372.704651460575
15154.860187919467
3909.7181112905796
569.2579648635418
176350.64991410912
106466.37224850639
Avg: 119141.72189571877