CSE 517 HW1

Zhanlin Liu

January 30, 2019

1 Smoothing

$$\begin{aligned} &p_1(w_i|w_{i-2},w_{i-1}) + p_2(w_i|w_{i-2},w_{i-1}) + p_3(w_i|w_{i-2},w_{i-1}) \\ &= p_{ML}(w_i|w_{i-2},w_{i-1}) + \frac{p_{ML}(w_i|w_{i-1})}{\sum_{w\in B(w_{i-2},w_{i-1})} p_{ML}(w|w_{i-1})} + \frac{p_{ML}(w_i)}{\sum_{w\in B(w_{i-1})} p_{ML}(w)}. \end{aligned} \tag{1}$$

$$p_{ML}(w_i|w_{i-2},w_{i-1}) = \frac{c(w_{i-2},w_{i-1},w_i)}{c(w_{i-2},w_{i-1})}, \tag{2}$$

where $c(w_{i-2},w_{i-1},w_i)$ is the count of the trigram (w_{i-2},w_{i-1},w_i) and $c(w_{i-2},w_{i-1})$ is the count of the bigram (w_{i-2},w_{i-1}).

$$\begin{aligned} \frac{p_{ML}(w_i|w_{i-1})}{\sum_{w\in B(w_{i-2},w_{i-1}))} p_{ML}(w|w_{i-1})} &= \frac{c(w_{i-1},w_i)}{c(w_{i-1})} / \sum_{w\in B(w_{i-2},w_{i-1}))} \frac{c(w_{i-1},w)}{c(w_{i-1})} \\ &= \frac{c(w_{i-1},w_i)}{\sum_{w\in B(w_{i-2},w_{i-1}))} c(w_{i-1},w)}. \end{aligned} \tag{3}$$

$$\begin{aligned} \frac{p_{ML}(w_i)}{\sum_{w\in B(w_{i-1})} p_{ML}(w)} &= \frac{c(w_i)}{c(\cdot)} / \sum_{w\in B(w_{i-1})} \frac{c(w)}{c(\cdot)} \\ &= \frac{c(w_i)}{\sum_{w\in B(w_{i-1}))} c(w)}. \end{aligned} \tag{4}$$

Therefore, above (1) can be rewritten as follows:

$$\begin{aligned} &p_1(w_i|w_{i-2},w_{i-1}) + p_2(w_i|w_{i-2},w_{i-1}) + p_3(w_i|w_{i-2},w_{i-1}) \\ &= \frac{c(w_{i-2},w_{i-1},w_i)}{c(w_{i-2},w_{i-1})} + \frac{c(w_{i-1},w_i)}{\sum_{w\in B(w_{i-2},w_{i-1}))} c(w_{i-1},w)} + \frac{c(w_i)}{\sum_{w\in B(w_{i-1}))} c(w)} \\ &= \frac{c(w_{i-2},w_{i-1},w_i)}{c(w_{i-2},w_{i-1})} + \frac{c(w_{i-1},w_i)}{\sum_w (c(w_{i-1},w) - c(w_{i-2},w_{i-1},w))} + \frac{c(w_i)}{\sum_w (c(w) - c(w_{i-1},w))} \\ &\geq \frac{c(w_{i-2},w_{i-1},w_i)}{c(w_{i-2},w_{i-1})} + \frac{c(w_{i-1},w_i)}{\sum_w c(w_{i-1},w)} + \frac{c(w_i)}{\sum_w c(w)} \\ &= p_{ML}(w|w_{i-2},w_{i-1}) + p_{ML}(w|w_{i-1}) + p_{ML}(w) \end{aligned} \tag{5}$$

From above equation, we can conclude it does not form a valid probability distribution as $\lambda_1 = \lambda_2 = \lambda_3 = 1$.

We can make it as one valid probability distribution by modifying p_1, p_2, and p_3 as follows:

$$
\begin{aligned}
p_1 &= p_{ML}(w_i|w_{i-2}, w_{i-1}) \\
p_2 &= (1 - \sum_{w \in A(w_{i-2}, w_{i-1})} p_{ml}(w|w_{i-1}, w_{i-2})(\sum_w p_{ml}(w|w_{i-1}))\frac{p_{ML}(w_i|w_{i-1})}{\sum_{w \in B(w_{i-2}, w_{i-1}))} p_{ML}(w|w_{i-1})} \\
p_3 &= (1 - \sum_{w \in A(w_{i-2}, w_{i-1})} p_{ml}(w|w_{i-1}, w_{i-2})(1 - \sum_w p_{ml}(w|w_{i-1}))\frac{p_{ML}(w_i)}{\sum_{w \in B(w_{i-1})} p_{ML}(w)}
\end{aligned}
\tag{6}
$$

This modification is generated from the Katz Back-off which gives weights regarding to the missing mass.

2 Language Models

2.1 Code descriptions

Three symbols including two STARTs ($< s >, < ss >$) and one END ($< /s >$) are added in the training, devoloping, and test set before building models. In order to handle out-of-vocabulary words, I first count the frequencies of words in the training set. Then words with low frequencies are turned as $< unk >$. After constructing the frequency dictionary for the unigram model, the words appeared in the developing set and test set but not in the training set are converted as $< unk >$. Based on the unigram frequency dictionary without low frequent words, unigram model, bigram model, and trigram model are further developed.

2.2 Results

As we can see in Table 1, it shows perplexity results for different datasets using different language models. We see the perplexity for bigram and trigram on the devloping set and test set is 0. It can be explained that there exists unseen bigram or trigram in the developing set which makes the loss as infinity.

Table 1: Perplexity results for datasets using different language models.

Model\Data	Train	Dev	Test
Unigram	3.3798	3.3323	3.3395
Bigram	2.4381	0	0
Trigram	1.6292	0	0

3 Smoothing

3.1 Hyper-parameters

3.1.1 Smoothing K

3.1.2 Linear interpolationλ

To prevent the case where the denominator for bigram and trigram is 0, $\frac{1}{corpus}$ is added on the denominator.

Table 2: Perplexity results for datasets using different K.

K\Data	Train	Dev
0.1	1.495	1.780
1	1.436	1.517
10	1.289	1.305
100	1.153	1.159
1000	1.068	1.070
10000	1.007	1.007
100000	1.001	1.001

Table 3: Perplexity results for datasets using different λ.

λ\Data	Train	Dev
(0.1, 0.6, 0.3)	1.513	2.334
(0.1, 0.3, 0.6)	1.570	2.365
(0.3, 0.3, 0.4)	1.378	2.354
(0.5, 0.3, 0.2)	1.284	2.410
(0.7, 0.2, 0.1)	1.232	2.547

3.1.3 Mixed hyper-parameters

When $K = 1000000$ $\lambda = (0.7, 0.2, 0.1)$, the developing set has the minimum perplexity. The perplexity for the test set is 1.01 using the hyper-parameters we optimize based on the developing set.

3.2 Half training

Using half dataset, the perplexity generally decreases. It can be easily explained as follows: as we decrease the training samples, the unique number of words would decrease as well. Therefore, the perplexity would decrease since the choice of words decreases.

3.3 Low frequency words

By converting words which appeared less than 5 times as UNK, the perplexity would decrease. By converting more words to UNK, the frequency of the bigrams or trigrams which involve UNK would increase. Therefore, we have more likelihood to predict $UNK|UNK$. Therefore, the perplexity would decrease since the choice of words decreases.

4 Language Model Classifier

References