CSE 517 HW1

Zhanlin Liu

January 31, 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}).

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

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

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 clearly see there exists overlapping between the three probabilities. Therefore, we conclude it does not form a valid probability distribution.

To make it as a valid probability distribution, we can take advantage of using the Katz-Backoff as follows:

$$
\begin{aligned}
p_1 &= \frac{c^\star(w_{i-2}, w_{i-1}, w_i)}{c(w_{i-2}, w_{i-1})}, w_i \in A(w_{i-2}, w_{i-1}), \\
p_2 &= \alpha(w_i)\lambda_1 \frac{p_{ML}(w_i|w_{i-1})}{\sum_{w \in B(w_{i-2}, w_{i-1})} p_{ML}(w|w_{i-1})}, w_i \in A(w_{i-1}) w_i \in B(w_{i-2}, w_{i-1}), \\
p_3 &= \alpha(w_i)\lambda_2 \frac{p_{ML}(w_i)}{\sum_{w \in B(w_{i-1})} p_{ML}(w)}, w_i \in B(w_{i-1}),
\end{aligned}
\tag{6}
$$

where

$$
\begin{aligned}
c^\star(w_{i-2}, w_{i-1}, w_i) &= c(w_{i-2}, w_{i-1}, w_i) - 0.5 \\
\alpha(w_i) &= 1 - \sum_{w_i \in A(w_{i-2}, w_{i-1})} \frac{c^\star(w_{i-2}, w_{i-1}, w_i)}{c(w_{i-2}, w_{i-1})}
\end{aligned}
\tag{7}
$$

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

$$
\begin{aligned}
p_1 &= p_{ML}(w_i|w_{i-2}, w_{i-1}) \\
p_2 &= (1 - \sum_{w \in A(w_{i-2}, w_{i-1})} p_{ml}(w|w_{i-1}, w_{i-2})(\sum_w p_{ml}(w|w_{i-1})) \frac{p_{ML}(w_i|w_{i-1})}{\sum_{w \in B(w_{i-2}, w_{i-1}))} p_{ML}(w|w_{i-1})} \\
p_3 &= (1 - \sum_{w \in A(w_{i-2}, w_{i-1})} p_{ml}(w|w_{i-1}, w_{i-2})(1 - \sum_w p_{ml}(w|w_{i-1})) \frac{p_{ML}(w_i)}{\sum_{w \in B(w_{i-1})} p_{ML}(w)}
\end{aligned}
\tag{8}
$$

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. In this report, the words which appear only 1 time in the training set are regarded as low frequent words.

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	868.10	765.90	769.37
Bigram	65.01	inf	inf
Trigram	7.03	inf	inf

3 Smoothing

3.1 Hyper-parameters

3.1.1 Smoothing K

Table 2: Perplexity results for datasets using different K.

K\Data	Train	Dev
0.01	183	3421
0.1	1225	5654
1	6769	10579

3.1.2 Linear interpolationλ

To prevent the case where the denominator for bigram and trigram is 0, $\frac{1}{|V|}$ is used for the probability if the bigram or unigram does not exist. V is the number of unique unigram.

Table 3: Perplexity results for datasets using different λ.

λ\Data	Train	Dev
(0.1, 0.6, 0.3)	27	267
(0.1, 0.3, 0.6)	34	297
(0.3, 0.3, 0.4)	16	282
(0.5, 0.3, 0.2)	11	305
(0.7, 0.2, 0.1)	9	387

3.1.3 Mixed hyper-parameters

The linear interpolation smoothing with $(\lambda_1, \lambda_2, \lambda_3) = (0.1, 0.6, 0.3)$ gives a smallest perplexity on the developing set. The perplexity on the test set is 268.

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

$$\begin{aligned} c^{\star} &= \arg\max_{c \in C} Pr(c|D) \\ &= \arg\max_{c \in C} Pr(D|c)Pr(c) \\ &= \arg\max_{c \in C} \{\prod_{i=1}^{N} Pr_c(w_i|w_1, w_2, \ldots, w_{i-1}, w_{i+1}, \ldots, w_n)\} Pr(c) \end{aligned} \tag{9}$$

Based on the above equation, we can sketch out how to do the text categorization using the language models. For each category, we can calculate the probability for having the sentences. Then we can classify the sentence as the category where it has the maximum probability.