INLP Assignment 1

REPORT

Ashmit Chamoli (2021101114)

Introduction

Perplexity of a Language Model is a lose measure of uncertainity of an LM when generating text. We use the following formula for calculating the perplexity of a sentence S:

$$PP[S] = P(w_1, ..., w_{|S|})^{\frac{-1}{|S|}}$$

where $w_1, ..., w_|S|$ are the words in the sentence S. For our N-gram model under the markov assumption, the formula reduces to the following form:

$$PP[S] = [\prod_{i=N}^{|S|} P(w_i|w_{i-1}, ..., w_{i-N+1}) \cdot P(w_1) \cdot P(w_2|w_1) \cdot ... \cdot P(w_{N-1}|w_{N-2}, ..., w_1)]^{\frac{-1}{|S|}}$$

We use 2 smoothing techniques for our N-gram model:

- Good Turing Smoothing
- Linear Interpolation

Smoothing Implementation

Good Turing Smoothing

According to the Good Turing estimate,

$$p_0 = \frac{N_1}{N}$$
$$p_r = \frac{r^*}{N}$$
$$r^* = (r+1) \cdot \frac{N_{r+1}}{N_r}$$

Where N_r is the number of species with frequency r and $N = \sum r \cdot N_r$ i.e. the total number of species.. Here, p_r is the estimate for the probability of seeing a species with frequency r. Therefore, p_0 gives us an estimate of the probability of seeing a species with frequency 0 i.e. unseen species.

Using these estimates, we can calculate the probability $P(w|w_1, w_2)$ as follows,

$$P(w|w_1, w_2) = \begin{cases} \frac{N_1}{(\sum r^* + N_1) \cdot (|V| - B)}, & w_1 w_2 w \notin Z \\ \frac{r^*}{\sum r^* + N_1}, & w_1 w_2 w \in Z \end{cases}$$

Where r is the frequency of 3-gram $w_1 w_2 w$ in the corpus, Z is the set of all 3-grams in the corpus, V is the vocabulary and $B = |\{w|w_1 w_2 w \in Z\}|$ i.e. B is the number of unique words that appear with $w_1 w_2$ in the corpus.

To calculate the estimate of r^* i.e. the smoothed probabilities, we perform the following steps,

1. Calculate Z_r from N_r as follows:
$$Z_r = \frac{N_r}{0.5 \cdot (t - q)}$$
and where q, r, and t are three consecutive subscripts with non-zero counts N_q, N_r, N_t. For the special case when r is 1, take q to be 0. In the opposite special case, when $r = r_{last}$ is the index of the last non-zero count, replace the divisor $\frac{1}{2}(t - q)$ with $r_{last} - q$ so $Z_{r_{last}} = \frac{N_{r_{last}}}{r_{last} - q}$.
2. Now fit a simple linear regression model to $log(Z_r)$ and $log(r)$, $log(Z_r) = a + b \cdot log(r)$.
3. Now we can estimate N_r as,
$$S(N_r) = exp(a + b \cdot log(r))$$

Perplexity Scores

For each of the 2 corpora provided, we construct a test set by selecting 1000 sentences at random and a train set containing the rest of the sentences. We calculate the perplexity of each sentence and report the average over all sentences for each language model.

LM 1: Tokenization + 3-gram LM + Good-Turing Smoothing

Pride and Prejudice

Train	90651.12
Test	12722.59

The high perplexity values on the test set indicate that the model is quite confused about it's prediction. The higher value of perplexity in the train set is because Good Turing smoothing assigns a very high probability to unseen n-grams which in turns results in probability of seen but low frequency n-grams to be extremely low. In the test set however, we see a lot of unseen n-grams for which the good turing model returns a very high probability.

Ulyess

Train	215313.48
Test	16327.17

Here similar trend is followed, except that the train perplexity is much higher than we see in Pride and Prejudice. This is because of a much larger vocabulary set and a larger dataset. This results in the probability of unseen n-grams to be even higher than was the case in Pride and Prejudice dataset.

LM 2: Tokenization + 3-gram LM + Linear Interpolation

Pride and Prejudice

Train	27.70
Test	813.61

Ulyess

Train	97.05
Test	2463.43

The perplexity in the for the train sets is quite low, indicating that the model is quite sure of it's prediction. On the test sets however, the overall perplexity is much lower than we see in Good Turing, meaning that the model is performing better in this case.

The perplexity scores are always higher for the Ulyess dataset as compared to the Pride and Prejudice dataset. This might be because of the richer vocabulary in the Ulyess dataset.

The best performance we achieve is by LM2 on the Pride and Prejudice dataset.