E1 246
Natural Language Understanding
Assignment 1 : Language Models

Monish Kumar Keswani
Department of Computer Science and Automation
monishkumar@iisc.ac.in

Abstract

A language model is designed on **Brown**(D1) corpus and **Gutenberg**(D2) corpus using N-grams model. Following two tasks are performed based on the model designed

- **1.** The dataset is divided into train, test and dev set.The training splits are D1-Train and D2-Train. The model is evaluated using the perplexity as the evaluation measure under the following scenarios
  - **S1 :** Train: D1-Train, Test: D1-Test
  - **S2 :** Train: D2-Train, Test: D2-Test
  - **S3 :** Train: D1-Train +D2-Train, Test: D1-Test
  - **S4 :** Train: D1-Train + D2-Train, Test: D2-Test
- **2.** Generate the sentence of 10 tokens using the model designed.

1 Unknown Word handling

The words that are seen in the training data may not appear in the Test data so for such words the probability will always be zero. There has to be some mapping so that such words do not get zero probability. So in the dictionary that we generate from the training data, we include a word called **UNKNOWN**. The word **UNKNOWN** . will have initial count as zero. So we have to map words from the training data to **UNKNOWN**. We can fix a threshold and all the counts below the threshold will be mapped to **UNKNOWN**. The threshold is increased gradually and the perplexity is calculated. It can be seen in the graph below that as the

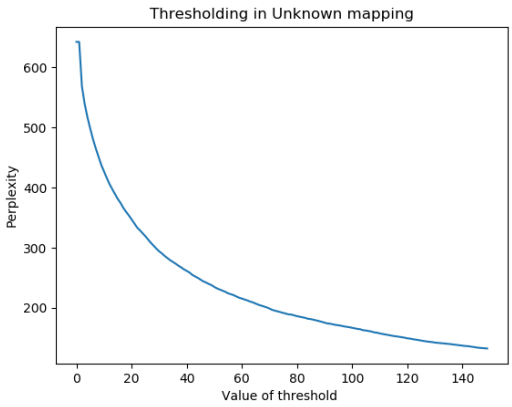

Figure 1:

threshold increases the perplexity decreases exponentially. This will cause over-fitting the training data. For the **Gutenberg**(D2) the number of words with count 1 are around 19000 out 40,000 vocabulary. So even if we set threshold as 1, it will remove significant vocabulary from the dictionary which is undesirable. So we select the top 1000 words with count 1 and map them to **UNKNOWN**.

2 Smoothing Techniques

2.1 Additive Smoothing in Unigram

For the unigram case we can use the UNKNOWN mapping as seen above. We can also use Laplace Smoothing.The graph of Additive-K smoothing is shown in Figure 2. It can be seen that the with increasing k from 0.1 to 1. It decreases initially and the shoots up. This means that the probability of **UNKNOWN** is increasing with increase in K.The UNKNOWN mapping technique is performing better in the unigram case hence we are using this technique for the unigram

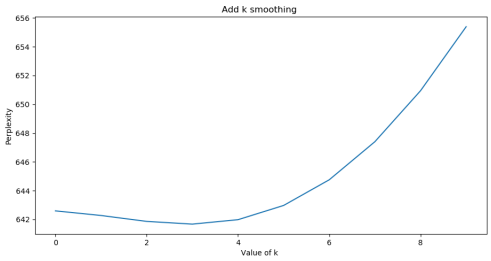

Figure 2:

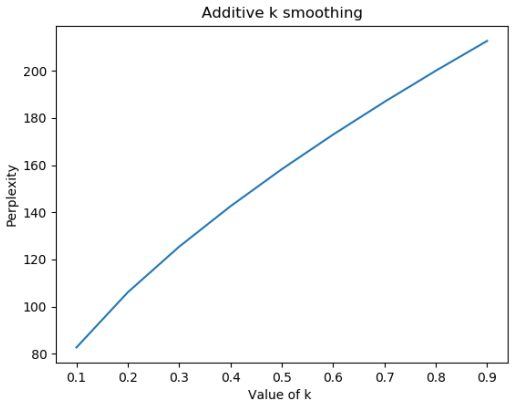

Figure 3:

3 Results

The following are the perplexity values based on the above models:

- S1:
  1. The perplexity for Linear Interpolation with trigram is 184.238
  2. The perplexity for Kneser Nay with bigram is 156.98

- S2:
  1. The perplexity for Linear Interpolation with trigram is 88.27
  2. The perplexity for Kneser Nay with bigram is 91.67

- S3:
  1. The perplexity for Linear Interpolation with trigram is 61.26
  2. The perplexity for Kneser Nay with bigram is 110.21

- S4:
  1. The perplexity for Linear Interpolation with trigram is 32.69
  2. The perplexity for Kneser Nay with bigram is 68.64

4 Sentence Generation

We generated sentences using Kneser nay bigram , that takes previous word into account. we pick a word from the probability distribution of the first word in the bigram and then generate the sentences.