REPORT Assignment 1 (Intro to NLP)

Average Perplexity Scores Language Models

Pride and Prejudice Corpus:

- LM1_train - 16.392301881461385
- LM1_test - 760.467166735325
- LM2_train - 24.422938752217785
- LM2_test - 330.05564682563565

Ulysses Corpus:

- LM3_train - 16.476070914738898
- LM3_test - 5970.782103537958
- LM4_train - 20.88825752493409
- LM4_test - 553.6206557343851

Analysing Different Language Models

1. **NGram Model (without any smooting technique used)**

- NGram Models calculate probablities simply on the basis of frequencies of those words in the given dataset, hence they are easier to work with and computationally efficient.
- They perform poorly for rare ngrams as they are simply frequency based.
- The major drawback is that for unseen ngram present in a given sentence, the model assigns the probablity as zero, leading to the probablity of occurenece of the entire sentence to be zero.
- In case of generation, using this model for different values of N the observations are as follows -
    - For N = 1, the model predicts the words with maximum frequency in the corpus (which mainly consists of stop words) as the next word. This is because without any given context, these words have the maximum probablity of occurence.
    - For N = 2, if the sentence already exists in the corpus the model predicts the next words which are fluent to the language but the probablities of all predicted words are very similar, leading to multiple possible next words with similar probablities.
    - For N = 3, the fluency and the difference between the probablities of predicted next word increases because of increased context for the task of prediction.
    - For N = 4 and above the data starts becoming sparse as there are only a few cases of occurence of a words after 3 or more context words and hence we say that the
    - From the observations on different values of N, we can say that. **"As the value of N increases, the accuracy of predicting the next word increases because of better discrimination between the probablities, but the larger N grams are rare leading to less reliability on the model."**
    - In Out-of-Data (OOD) scenario, the Ngram model randomly chooses the next word with weight give to frequency of its occurence in the data.
    - Example of predicted words when N = 2:
        - input sentence: Mr. Bennet replied that he
        - had 0.13259668508287292

        - was 0.1132596685082873
        - is 0.04696132596685083
    - Example of predicted words when N = 3:
        - input sentence: Mr. Bennet replied that he
        - had 0.21176470588235294
        - was 0.1588235294117647
        - should 0.09411764705882353
    - Example of predicted words when N = 4:
        - input sentence: Mr. Bennet replied that he
        - had 0.5
        - was 0.25
        - did 0.25
    - Example of predicted words when N = 5:
        - input sentence: Mr. Bennet replied that he
        - had 1.0

For handling the problem of sparse data and finding unseen probablities, two smoothing techniques were implemented

2. **Good Turing Smoothing**

- It provides a simple estimate probablities for the objects not seen and makes sure that estimate probablites for the observed objects are consistent with the total probablity assigned to the unseen objects, using Linear Regression.
- The probablities are calculated with the assumption that 'Word classes with similar frequency counts can be treated similarly.' and therefore the probablities of lower frequency words are calculated with the help of higher frequency words.
- For observed trigrams the Probablity is calculated using count*[trigram]/sum(count*[context]), for observed context the probablity is calculated as Nr1/sum(count*[context]) and for unkown context a very low probablity of 1/V is assigned.
- For training data, Good Turing is more accurate in comparision to other models, but for testing data its performance worsens, because for the unseen cases, good turing assigns a fixed and low probablity.
- For generation using Good Turing, if the context words(last 2 words in the input sentence) are present then the model accurately predicts the next word, but if the context words are not present in the data then the model outputs nothing.
    - Example of predicted word -
        - input sentence: Mr. Bennet replied that he
        - had 0.2030256686191218
        - was 0.1501831134806875
        - should 0.08573804153942463

3. **Linear Interpolation**

- In Interpolation, to handle trigrams, which do not exist in the dataset, we mix the probablity estimates from all the ngrams (in this case trigrams, bigrams and unigrams), in other terms use less context for prediction.
- The probablities are calculated as follows:

$$\begin{aligned}\hat{P}(w_n|w_{n-2}w_{n-1}) &= \lambda_1 P(w_n) \\ &\quad + \lambda_2 P(w_n|w_{n-1}) \\ &\quad + \lambda_3 P(w_n|w_{n-2}w_{n-1})\end{aligned}$$

- The values of l1, l2 and l3 are calculated using the following algorithm:

```
set λ1 = λ2 = λ3 = 0
foreach trigram t1,t2,t3 with f(t1,t2,t3) > 0
    depending on the maximum of the following three values:
        case (f(t1,t2,t3)−1)/(f(t1,t2)−1):  increment λ3 by f(t1,t2,t3)
        case (f(t2,t3)−1)/(f(t2)−1):  increment λ2 by f(t1,t2,t3)
        case (f(t3)−1)/(N−1):  increment λ1 by f(t1,t2,t3)
    end
end
normalize λ1,λ2,λ3
```

- Since, Interpolation mixes various Ngram models for probablity calculation, it works better in cases when the complete context was unseen, leading to better perplexity values in case of test data in comparision to Good Turing Method.
- For generation task, even if the complete context (in this case, last two words of the input sentence) is not present in the data, we can still predict the next word using a shorter context leading to better prediction in case of unseen contexts.
    - Example of Predicted Word:
        - input sentence: Mr. Bennet replied that he
        - had 0.1082560684855081
        - was 0.08890029565774481
        - is 0.03523177932007003

Analysing the difference in Preplexities for both Datasets

- In general, it was observed that the perplexity of large sentences was more than that of shorter sentence. Since the Ulysses Corpus contained a lot of large sentences, the perplexities for the Pride and Prejudice Corpus were found to be better because of shorter sentences.

The choice of Smoothing method to be used depends on a lot of factors such as the length of sentences in the corpus, datasize, sparsity of the data etc.