NLU : Assignement - I

anirbanb@iisc.ac.in

1 Task - I

1.1 Description

In this task, given two datasets *D1:Brown Corpus* and *D2:Gutenberg Corpus*, the task is to experiment with language models and build the best one in the given four settings. I have implemented four different models namely **unigram**, **bigram**, **trigram** and **mixgram** (which is basically the interpolated model of the earlier three). I have tried *Add-One* as well as*Good-Turing* as smoothing techniques.

For all four different settings given in the problem, trained the models first and then calculated perplexity based on test data. The final model have been chosen based on the perplexity values obtained for the various settings. It turns out that the interpoplation/mixgram model along with good turing smoothing gives best results i.e. lowest perplexity value among all the models (clearly seen from the results section below).

Now once I finalized my model, I tuned the lambda parameters which are hyperparameters for the interpolation model. I tried with varoius settings of hyperparameters and finally took the one which gave best perplexity.

The division of train and test set size I have used is as follows: *Train set - 98% and Test set - 2%*

1.2 Results

The following four graphs, one for each of the four settings mentioned in the problem, shows the comparison of perplexity measures of all the four models.

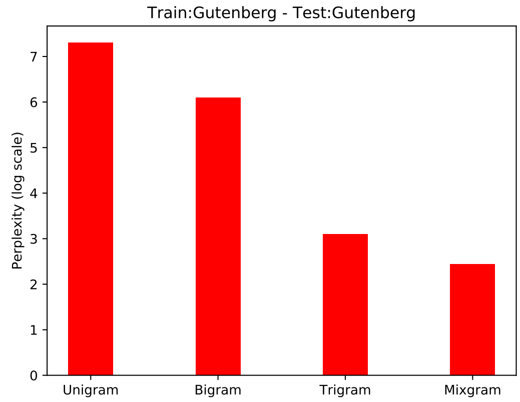


Figure 1: S1: Train: D1-Train, Test: D1-Test

All the bar graphs gives measure of perplexity values for unigram, bigram, trigram and mixgram/interpolation model for different settings. The smallest bar height implies best model.

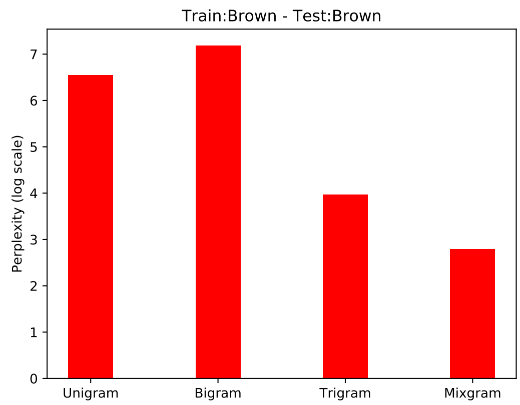


Figure 2: S2: Train: D2-Train, Test: D2-Test

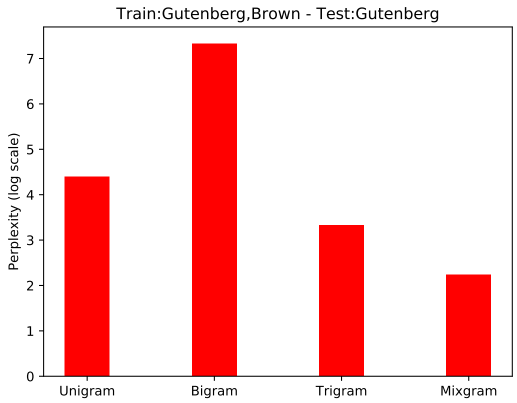


Figure 3: S3: Train: D1-Train + D2-Train, Test: D1-Test

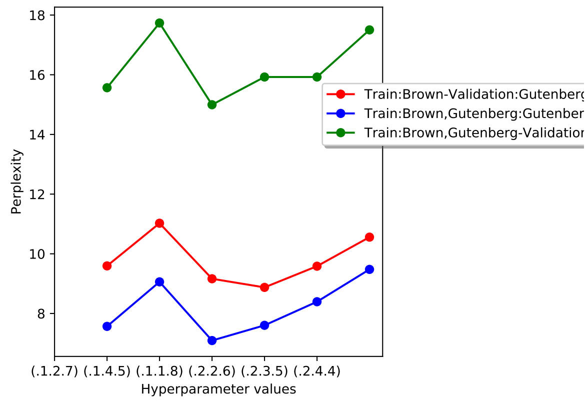
Figure 4: S4: Train: D1-Train + D2-Train, Test: D2-Test

1.3 Discussion

It can be clearly seen from all the graphs that the interpolation (a.k.a mixgram) model with Good Turing as smoothing technique gives lowest perplexity, hence best performance. Therefore, I fix this model as my final LM. Now this model's perplexity depends on three hyperparameters - λ_1, λ_2, λ_3, which are wights for unigram, bigram and trigram respectively. The trend is as we go from unigram to trigram, perplexity is getting less as we are adding more context. Now the interpolation model is performing best because in that case we are taking into account of all three models.

Now I need to tune these hyperparameters. The follwoing section shows results of hyperparaneter tuning.

1.4 Results : Hyperparameter Tuning



1.4.1 Final Model

Based on the above plots, it can be easily seen that for hyperparameter setting $\lambda_1 = 0.2$, $\lambda_2 = 0.2$, $\lambda_3 = 0.6$, this models performs best. Hence our final model is interpolation model with this hyperparameter setting. We use this model to generate the sentence for task 2.

2 Task - II

2.1 Description

Using the best model obtained from task-I, this task requires to generate a sentence of 10 tokens based on the trained model. Sentence generation task is done in such a way that it starts with the start sentence token. A next word is generated based on taking all the words with matching context and then selecting one word randomly. The next word is then generated in the same way, only the context now has been changed with the lastly added word comes into new context. Once 10 tokens are generated this process stops.

Some example sentences generated by this model are - *" I am sure I should be the LORD* , , *" I am sure I should not be afraid of* etc.