NLU Assignment-1 Report

Vipul Kumar Rathore - 14754
rathorevipul28@gmail.com

1 Model

- **Language model** - N-gram Model used with **N = 2 (Bigram model)**. Easy to implement and giving good results.
- **Training Set** - 80 percent of corpus size for each of the 4 settings
- **DevSet** - used to tune the value of N but later removed after tuning
- **TestSet** - 20 percent of corpus size for each of the 4 settings
- **Smoothing** - Stupid Back-off technique: If N-gram not present then calculate the probability of (N-1)-gram, and so on till unigram.If unigram is also not present, then the probability = (1/(vocab. size)), which is derived from the add-1 smoothing for the case when a unigram is not present in the training corpus.
- **Evaluation Metric** - Perplexity measure

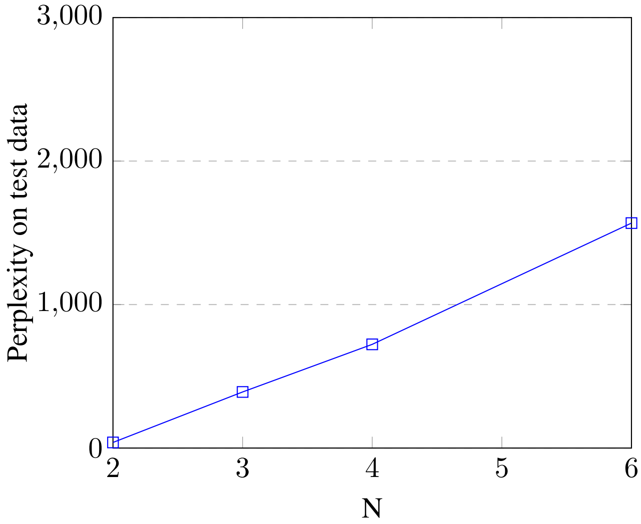
2 Model parameter Tuning

I tuned my model only for first setting and then used the optimal hyperparameters obtained from first setting as hyperparameters for all the 4 settings.

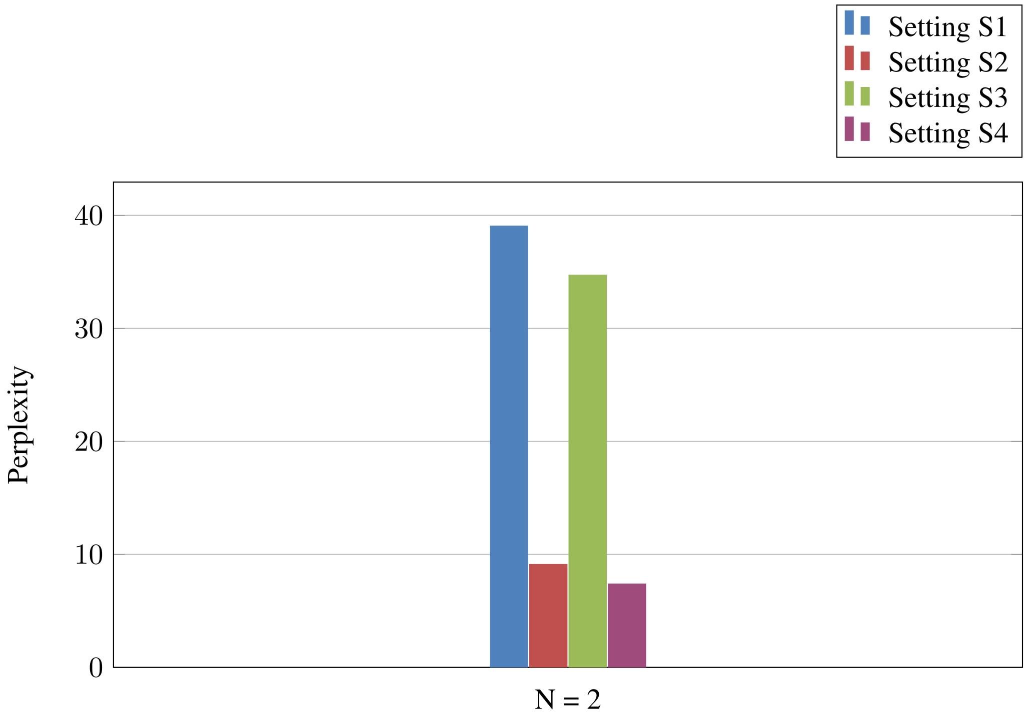
- **Value of N** - I tried N = 2,3,4,... so on and the optimal value of N giving lowest perplexity on dev. set (10 percent) is N=2.
- **Smoothing technique** - I tried the smoothing techniques like add-1 smoothing, Kneser-Ney smoothing and stupid back-off. The technique giving the best perplexity results on dev. set is stupid backoff which gives around 10 times better perplexity than kneser ney and around 25 times better than add-1 (laplace) smoothing, all other hyperparameter settings remaining constant.

3 Results and Plots

- The plot of perplexity on test data for various values of N for setting S1 is given below.



- For our model i.e. N = 2, the plot of perplexity with the 4 settings is as follows.



S1	S2	S3	S4
39.043	9.108	34.70	7.368

- Thus the best results are obtained on **setting 4** for **N=2** and the perplexity value is **7.368**.

  Github Code link - https://github.com/rathorevipul28/NLU_Assignment1