1. How did you select the final sets of parameters (N)? What are the values?

The procedure that we followed for getting the optimal values of N is as follows:

The first step is **taking each training file in the directory one by one and building a N-gram** (unigram, bigram model and trigram) model on it. Since we had 55 training files in the directory, therefore after the first step we have 55 Unigram Models, 55 Bigram Models and 55 Trigram models each built on the respective training file.

Then we move to the evaluation stage. Here, we **go through each file in the dev directory one by one** and we **use ALL the models built above to assign perplexity to that dev file.** Next we sort these perplexities and hence **find which Ngram model (and the corresponding training file that was used to build it) gave the least perplexity** on the contents of that particular dev file. This will be the final prediction for that dev file and this is written to the output file.

The above two steps are repeated for the other 2 types of models: Laplace Smoothing and Interpolation after making the necessary changes in probability calculation for each type of model.

Regarding the values of N, we found that unigrams (N=1) models were giving the least perplexity value for all the three models. But **for some test files** like “udhr-cjk.txt.dev”, “udhr-kin.txt.dev” and “udhr-nba.txt.dev”, **the optimal value was N=2 in case of the Unsmoothed model. Trigrams usually gave worse performance than Unigrams and Bigrams.** One reason we think this is because of the higher number of trigrams that appear in the dev text but were not present in the training text.

2. Which model performed the best? Discuss the relative performance of the smoothing variants and n-gram settings.

With respect to the comparison of the three types of model variants, we found that **the perplexity values of unsmoothed and laplace variants were similar** for all the languages. The perplexity values we predicted usually **ranged between 5 to 20 for both types of models.** On the other hand, we found that the i**nterpolated model was much better as the perplexity predictions for it ranged between 1 to 3.**

For e.g. Consider the test file “udhr-eng.txt.dev”. Our laplace model predicted a perplexity of **17.30776** whereas the unsmoothed model predicted perplexity as **17.28479** for the same file. The linearly interpolated model was much better as it gave a perplexity prediction of just **2.60795** for the same dev file.