- train.txt: data for training your language model.
- dev.txt: data for developing and optimizing your language model.
- test.txt: data for only evaluating your language model.

You will first build vocabulary with the training data, and then build your own unigram, bigram, and trigram language models with some smoothing methods.

2.1 Building Vocabulary

You will download and preprocess the tokenized training data to build the vocabulary. To handle out-of-vocabulary (OOV) words, you will convert tokens that occur less than three times in the training data into a special unknown token `<UNK>`. You should also add start-of-sentence tokens `<s>` and end-of-sentence `</s>` tokens.

Please show the vocabulary size and discuss the number of parameters of n-gram models.

2.2 N-gram Language Modeling

After preparing your vocabulary, you are expected to build bigram and unigram language models and report their perplexity on the training set, and dev set. Please discuss your experimental results. If you encounter any problems, please analyze them and explain why.

2.3 Smoothing

In this section, you will implement two smoothing methods to improve your language models.

2.3.1 Add-one (Laplace) smoothing

Please improve your bigram language model with add-one smoothing. Report its perplexity on the training set and dev set. Briefly discuss the differences in results between this and the bi-gram model you built in Section 2.2.

2.3.2 Add-k smoothing

One alternative to add-one smoothing is to move a bit less of the probability mass from the seen to the unseen events. Instead of adding 1 to each count, we add a fractional count k (.5? .05? .01?). This algorithm is therefore called add-k smoothing, as shown here in the bigram case:

$$P^*_{\text{Add-k}}(w_i|w_{i-1}) = \frac{C(w_{i-1}\ w_i) + k}{C(w_i) + kV} \tag{3}$$

Please optimize the perplexity on the dev set by trying different k (no more than three times). Report its perplexity on the training set and dev set. Briefly discuss the differences in results between this and the bi-gram model with add-one smoothing.

2.3.3 Linear Interpolation

Please implement linear interpolation smoothing between unigram, bigram, and trigram models. Report its perplexity on the training set and dev set. Please optimize on a dev set by trying different hyperparameter sets. Finally, report the perplexity on the test set with the best hyperparameter set you get. Briefly discuss the results.

Optimization. So far, we manually choose the hyperparameter that maximizes the perplexity of the dev set and evaluates it on the test set. There are various ways to find this optimal set of hyperparameters. Do you know any other learning algorithms? Please give an example.