3 Tuning an n-gram language model (20 points)

This Colab notebook contains starter code for this problem.

When we covered n-gram models, you learned about MAXIMUM-LIKELIHOOD ESTIMATION and ADDITIVE SMOOTHING for a bigram model. Maximum-likelihood estimation is equivalent to RELATIVE FREQUENCY ESTIMATION:

$$\widehat{P}_{RFE}(w_i|w_{i-1}) = \frac{\text{Count}(w_{i-1}w_i)}{\text{Count}(w_{i-1})}$$

In ADDITIVE SMOOTHING, a pseudo-count of α is added to each n-gram count, so that the bigram probability of w_i given w_{i-1} is estimated as

$$\widehat{P}_{\alpha}(w_i|w_{i-1}) = \frac{\text{Count}(w_{i-1}w_i) + \alpha}{\text{Count}(w_{i-1}) + \alpha V}$$

where V is the vocabulary size.

Task: consider the following toy datasets, similar to those appearing in recitation 4:

Training set	**Validation set**	**Test set**
`dogs chase cats`	`the cats meow`	`cats meow`
`dogs bark`	`the dogs bark`	`dogs chase the birds`
`cats meow`	`the dogs chase the cats`	
`dogs chase birds`		
`birds chase cats`		
`dogs chase the cats`		
`the birds chirp`		

1. Implement a bigram language model with additive smoothing, trained on the training set and with the smoothing parameter α optimized to mimimize the held-out perplexity of the validation set (make sure that you include beginning-of-sentence and end-of-sentence tokens in your representation of each sentence, and that you include these tokens appropriately in your perplexity computations). In choosing which possible values of α to consider, you can use any of a variety of methods; the simplest is grid search, but you can use another method if you prefer.

2. Plot the validation-set perplexity against the test-set perplexity, considering a range of α values (think carefully about what range you need to consider in order to fully understand how the validation and test sets test generalization from the training set). What value of α worked the best for the validation set? Was it the same that would have worked best for the test set?

Want to take this farther? (BONUS 5 points) Once you've done the above you've completed the assignment, but for further learning you can try tuning α for a larger-scale dataset, such as Wikitext-2 or Wikitext-103 (training sets of about 2 million words and 103 million words respectively, which are still small by contemporary NLP standards!), This will also give you an opportunity to see how well your implementation scales time- and memory-wise as the dataset gets larger, which is often very important for computational work on language.

Notes and suggestions

Background note: in machine-learning terminology, choosing the value of α based on the held-out perplexity of the validation set is an example of HYPERPARAMETER TUNING. In general, you don't get to look at the test-set performance for all the different choices of the hyperparameters; you have to make your hyperparameter commitments on the basis of exploration against the validation set, and then look at performance on the test set at the end. In the present exercise, We are asking you to look at both together strictly for pedagogical purposes.

Open versus closed vocabulary language modeling: if you try to scale up to a larger dataset like suggested above, you will have to deal with the question of how to define the vocabulary size V. In the general case, there may be words in your validation and/or test sets that do not appear in your training set, so you cannot just set V to the number of distinct words that appear in the training set, or you will wind up with an improper probability distribution (why?). A couple of alternative options include:

- "Peeking" at the validation and test sets and defining the vocabulary as including at least the union of all words that appear in training, validation, and test sets; this is what is known as CLOSED VOCABULARY language modeling.
- Defining a vocabulary that does not necessarily include all words that appear in the validation and/or test sets, together with an "unkification" function that maps words that do not appear in this vocabulary to one or more UNKNOWN WORD categories (when there is only one unknown word category, it's traditionally denoted with `<UNK>`). You then convert your training, validation, and test datasets using this unkification function, and all perplexity evaluations involve these converted dataset. This is OPEN VOCABULARY language modeling.

These issues are covered in some more detail in SLP3 section 3.3.1. Note that you don't have to worry about any of this for the toy dataset for the present problem, because every word appearing in the validation and test sets also appears in the training set.