Currently the (cl n-grams) library contains only one procedure: n-gram-tallier, which reads in text through a given input port, tokenizes it, and constructs and returns a hash table in which the keys are N-grams (sequences of N successive tokens) and the values are positive integers indicating how many occurrences of the corresponding N-grams were encountered in the text. The value of N can be any positive integer and is supplied by the caller.

For a word-prediction model, we’d prefer to have the same data in a different structure, one that would make it easy to take any sequence of N − 1 tokens and look up the relative frequencies of the tokens that can follow it. The object of today’s lab is to build that data structure.

Because the data are still going to be sparse in most cases, since most sequences of N − 1 tokens won’t occur at all in the text when N ≥ 3, it’s once again better to use a hash table to associate keys and values in this case. Considering the structure of the word-prediction task, I suggest that we using (N − 1)-grams as keys and association lists (that is, lists of pairs in which no two cars are the same) as values. In these association lists, the car of each pair will be a token that has been observed to follow the N − 1 tokens in the key, and the cdr will be the relative frequency of the car in that context. We’ll use equation (3.12) from the third edition of our textbook (which is equation (4.15) in the second edition) to compute the relative frequencies.

1. As an intermediate step, make a copy of the (cl n-grams) library and add to it a new helper procedure (not for export) called move-end-tokens-into-values. This new procedure should take a hash table structured like the ones that n-gram-counts produces and constructs and returns a new hash table with a structure similar to the one we ultimately want, but with raw tallies instead of relative frequencies. The keys for the new hash table should be (N − 1)-grams, and each of the values should be an association list in which the cars are tokens that were observed to follow the corresponding key and the cdrs are the tallies of the number of occurrences of that observation.

For instance, when N = 2 and the text is the three-line “I am Sam” sequence from the textbook, the n-gram-counts procedure already yields a hash table in which the key ("I" "am") is associated with the value 2 and the key ("I" "do") is associated with the value 1. Given this table, the move-end-tokens-into-values procedure should build a hash table in which the key ("I") is associated with the value (("do" . 1) ("am" . 2)).

(Hint: Use hash-table-walk to traverse the given hash table.)

2. Design and implement another helper procedure, sum-of-cdrs, for the (cl n-grams) library. This one should take an association list, similar to the values in the hash tables produced by move-end-tokens-into-values, and compute and return the sum of its cdrs. (This value is the denominator $C(w_{n-N+1}^{n-1})$ of the fraction on the right-hand side of the equation cited above.)

$w_{n-N+1}^{n-1}$ is one of the keys in the hash table produced by move-end-tokens-into-values, and we’re computing the total number of occurrence of that (N − 1)-gram in the original text.

3. Design and implement another helper procedure, normalize-cdrs, that takes an association list, similar to the values in the hash tables produced by move-end-tokens-into-values, and returns a similar association list in which each of the cdrs has been divided by the sum of all the cdrs. (Basically, we’re calculating the values of the fraction on the right-hand side of the cited equation—the relative frequency of each token $w_n$ given the context provided by the sequence $w_{n-N+1}^{n-1}$ containing the N − 1 preceding tokens.)
4. Combine the three helper procedures and the existing \texttt{n-gram-tallier} procedure to implement the \texttt{n-gram-relative-frequencies} procedure envisioned in the introduction to this lab. It should take as arguments a positive integer \( n \) and an input port \texttt{source} and return a hash table in which the keys are the \((N - 1)\)-grams occurring in the text and each value is an association list in which the cars are tokens and the cdrs are the relative frequencies of those tokens when they follow the tokens in the key.

(Hint: You should find the \texttt{hash-table-keys} procedure useful in organizing the computation.)

5. (Optional exercise.) Write a test procedure that takes a hash table produced by your \texttt{n-gram-relative-frequencies} procedure and checks to make sure that the sum of the relative frequencies in each value that occurs in the hash table is equal to 1. (If they are going to be interpreted as probabilities of mutually exclusive and jointly exhaustive outcomes of an event, failing to meet this condition would show that there is something seriously wrong with the code.)