EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

PHILOSOPHISCHE
FAKULTÄT
Seminar für Sprachwissenschaft

Data Science for Linguists
Summer 2025
Johannes Dellert

Assignment 01:
Token Frequencies in a First Jupyter Notebook

handed out: 23 April, 14:00
to be submitted by: 30 April, 12:00

The purpose of this first exercise sheet is to solidify elementary Python skills that will be needed throughout this course, and establishing the workflow for the graded assignments. You will be working with the file `sq-sample.txt`, a sample of 100,000 alphabetically ordered sentences from an Albanian webcrawl, in order to build frequency lists of the word forms that would belong e.g. on vocabulary lists.

Task 1: Basic Setup and First Steps

Set up IPython and Jupyter on your machine, following instructions for your operating system. Create a new Jupyter notebook, and play around to familiarize yourself with the workflow. Create at least two code cells and two Markdown cells. Extract the full inventory of Unicode characters occurring in the contents of this file. You will find the file is quite polluted with material in other writing systems, so it seems best to instead create a "positive filter" for purely Albanian words, which must consist of only the following letters (uppercase or lowercase): *a, b, c, ç, d, e, ë, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, x, y, z.* Implement the filter as a function `is_albanian` which returns a `True` if it matches this filter, and `False` otherwise.

Task 2: Loading and Preprocessing Corpus Data

Read in the contents of the file, scramble the sentences (important!) and tokenize them by splitting it at whitespaces and linebreaks, turning all tokens to lowercase, and removing all the important punctuation symbols (commas, full stops, parentheses, ...) and digits. Using `is_albanian`, store the purely Albanian tokens in a list. You should end up with a list of somewhere between 1.5 million and 2 million tokens.

Task 3: Populating Data Structures

a) Separate the tokenized text into two subcorpora of roughly equal size. Convert your lists of tokens into two `Counters` (instances of the relevant class from the `collections` package) of token frequencies. Print out the top-50 most common word forms in both halves. Is there any difference between subcorpora?
b) Write a function which goes through a token counter by order of frequency, and keeps track of the cumulative probability of all tokens encountered so far in order to split them into quantiles. A named argument `k` should make it possible to configure the size of the quantiles (`k=2` for quartiles, `k=100` for percentiles, ...). Use `defaultdict` (again from the `collections` package) to return a map from quantiles (integer keys: `1` for first quantile, `2` for second quantile, ...) to a set of tokens placed in that quantile. Run the function with `k=10` on both subcorpora, and store the results.

Task 4: Answering Questions about the Data

a) Use set operations on the stored results in order to print out a table of overlap sizes between the subsets placed into the deciles on each of your two subcorpora. Use the table to determine how many forms we would need to know in order to understand 20%, 50%, 70%, 80%, and 90% of tokens across both corpora.
b) If vocabulary learning was maximally efficient, roughly how much of the corpus would you be able to read using typical lexical coverages of the CEFR levels A1 (625 words), A2 (1250 words), B1 (2500 words), and B2 (5000 words)? Do the results seem plausible? If you can spot a problem, what could it be?

Task 5: Exporting Your Notebook

Clean up your code, add Markdown cells with your answers to the questions. Export your notebook as a PDF, and upload it to Moodle for grading and feedback.