Statistical Analysis of Corpus Data with R — Exercise Sheet for Unit #5 —

1. Type richness of German NP and PP constructions

Although the notions of productivity and type richness have been used mostly in morphology, similar ideas can also be applied to the study of syntax. The *zipfR* package comes with data sets reporting frequency information about the "expansions" of German NP and PP phrases in the German TIGER treebank (see `?TigerNP` for more information).

Import the German NP and PP data sets. For the data set with more tokens, compute a binomially interpolated vocabulary growth curve (VGC) using `vgc.interp()`. For the smaller one, estimate an LNRE model (pick your favorite model, or try them all) and use it to compute an expected VGC up to the size of the larger data set with `lnre.vgc()`. Plot the interpolated and extrapolated VGCs, and determine which of the two constructions appears to be more productive (in terms of type growth).

2. Data lost with cut-off points

Before you tackle this part of the exercise, read the second case study in the *zipfR* tutorial.

It is common, in collocation studies and similar work, to discard bigrams (i.e., sequences of two words) below a certain occurrence threshold, typically $f < 2$ (discarding bigrams that occur once, the *hapax legomena*) and $f < 5$ (discarding bigrams that occur 4 times or less). In this exercise, we try to assess the effect that such cuts have on the proportion of types considered, on the basis of a sample from a relatively small corpus.

First, load the file `bigrams.100k.tfl` (containing bigrams extracted from the first 100,000 tokens of the Brown corpus) as a *zipfR* type frequency list with `read.tfl()`, and generate a frequency spectrum from this with `tfl2spc()`.

What proportion of bigram types occur only once? What proportion of types occur 4 times or less?

Now, suppose that we want to use our data to estimate the proportion of bigram types that would be lost by using the same two frequency cut-offs if we had a 1 million word corpus. Fit a LNRE model to the observed frequency spectrum, then compute $\mathrm{E}[V]$ and $\mathrm{E}[V_m]$ for a sample size of $N = 10^6$. You can either use `lnre.spc()` for this purpose, or obtain the necessary expectations directly with `EV()` and `EVm()`.

3. Reliability of the fitted model

In the last part of this exercise, you will use parametric bootstrapping to determine how reliable the estimated model parameters are. For our analysis of type richness, the slope parameter α and the population vocabulary size S are of particular importance; the goodness-of-fit statistic X^2 is also useful because indicates suboptimal parameter estimates.

Train an fZM model on the German NP data (`TigerNP.spc`). Apply `lnre.bootstrap` to generate 100 samples[1] from this model, fit a new fZM model to each sample, and extract the values of α, S and X^2. Combine the bootstrapped values into a data frame with 100 rows.

Use your knowledge about descriptive statistics from Unit 3a to describe and visualize the distribution of the three values. How do you interpret your observations?

Would we also be able to determine type richness based on a much smaller treebank? Apply the bootstrapping procedure to 10% of the original sample size ($N \approx 11{,}000$). Then compare the distribution of α, S and X^2 for the small samples with the previous results.

Bootstrapping can also give an indication how reliable it is to use LNRE models for the extrapolation of vocabulary growth curves. Re-run `lnre.bootstrap()` for a sample size of $N \approx 11{,}000$, but collect the extrapolated values $\mathrm{E}[V]$ and $\mathrm{E}[V_1]$ for the full data size (i.e. `N(TigerNP.spc)`). You will have to design a suitable callback function based on the methods `EV()` and `EVm()` in order to do so. Compare the distribution of the expected values to the actual values of V and V_1 for the German NP data.

[1]of the same size as the NP data