Assignment 9
Chris Potts, Ling 278: Programming for Linguists, Fall 2018
Distributed Nov 26; due Dec 5, by 11:59 pm

Logistics

Note the unusual due date: a Wednesday, after class. This gives you a full week plus our two project-work class meetings to complete the MVP. After that, you have another week before the final project is due (Dec 12, 6:30 pm).

If you turn this assignment in early, I’ll do my best to get comments to you quickly. This means that you can figure out what schedule is best for you and your project. For example, if your MVP is easy to obtain, then you might turn it in early to get feedback, and then you’ll be able to do more for the final submission.

If your submission consists of multiple files, please compress them using ZIP or GZIP (ZIP is the default compression on Macs.)

Requirements

In software engineering, a minimum viable product (MVP) is one that has “just those features that allow the product to be deployed, and no more” (http://en.wikipedia.org/wiki/Minimum_viable_product). This concept helps define the planning process, by pushing the designers to figure out what the core of their product really is, and it helps with implementation, by working to ensure that the code does not grow too complex too fast. It’s also meant to create a fast track to people (you) actually using the code, so that you get a realistic sense for what you need it to do.

Your task for this assignment is to define and submit an MVP for your final project. The submission should take the following form:

i. A notebook containing the following:
   a. A description of the central goal of your project. This can be an adaptation of your answer to problem 3 of assignment 7, but it should have the MVP concept in mind, in that it should articulate what the MVP does if it is more limited that the full project goal.
   b. A description of what your code does at present.
   c. A brief overview of how your code works — how the specific functions fit together, how they interact with the data, what they produce, and so forth.

ii. Any supporting modules – code that you wrote that is not embedded in the notebook. (It is fine at this stage to have all the code in the notebook, though I think you will want to factor much of it out into separate modules eventually.)

iii. A separate file called requirements.txt listing any packages your code requires that do not come with Anaconda. (If you had to install it separately, include it in this list.) Please include installation instructions.

It’s up to you to define what the MVP for your project is. I can only say that you want to keep it simple, but you also want it to be useful. The final project submission will improve and extend this code base, but you needn’t have extensions in mind right now, as that might compromise the MVP.