Kavi Mehta
Greta Huang

Spring 2018 URAP Write-Up

For this semester, we primarily worked on developing a collocation finder in C++. For the first few weeks, Greta worked on getting up to speed with the project at large and the linguistic approach (namely, learning about Metaphor and reading *Don't Think of an Elephant*), and Kavi worked on adding to and polishing the frames table from the Fall 2017 term.

Developing a collocation finder is an important and relevant piece of the long-term goal — that goal being to use machine learning to classify corpus as more liberal or conservative on the basis of the metaphors used. To start this process, we need to identify the metaphors commonly used in text currently, and one approach to doing this is to pull out collocations from text and manually identify which are prominent metaphors for both Nurturant Parent and Strict Father.

To begin developing a collocation finder, we initially found existing ones and tested them. We first tried the collocation finder included in Stanford's NLTK, which worked but was slow. Then, Dekai had us read Frank Smadja's paper on collocation finders, which formed the basis for his program Xtract, and led to a more recent Java implementation called JXtract. JXtract was also slow.

To create a faster collocation finder, we decided to recreate Xtract in C++ by mostly translating over from JXtract. This undertaking occupied the majority of the semester, and was particularly challenging as neither of us have worked with C++ before. After lots of translating, we got to the point of compiling and encountered bugs (expected). We fixed those bugs and encountered more bugs (expected). Repeat a few more times. We are now at the point of having only a few bugs left (hopefully the last ones!!) that appear to be due to the fact that we are not using pointers in our data structures, as is conventional in C programming.

Moving forward, we would like to finish the collocation finder and use it on a corpus of poilitical text to see what frames we can pull out. This will be valuable data and classifying phrases as either Nurturant Parent or Strict Father will be a great task for students trying to get started with the project.