Andrew Hoeft
CS 410 Fall 2020: Final Project Proposal

Team Members: Andrew Hoeft (arhoeft2)

Chosen Option: 2 – Improving a System
Chosen System: 2.1 – MeTA Toolkit
Chosen Subtopics: Enhancing MeTA and Metapy Usability

Datasets, Algorithms, Techniques:

No datasets are required to improve usability per se; the datasets from MP2 will be used for feature verification once complete.

No algorithms are required to improve usability per se; using the datasets from MP2 to verify functionality, I will also look for opportunities for local optimization that may result in applying some (hitherto unknown) algorithms.

For techniques, I intend to apply test-driven development to verify that the changes imparted with upgrading Python libraries does not change the underlying behavior of the MeTA library.

Improving Performance:

Testing for basic functionality (compilation, importation, execution) without error will be the first metric for successful improvement. If there are fundamental architecture changes that Python 3.9 imparts on the library, I will choose a set of queries provided in MP2 and perform and evaluate a time study between equivalent Conda runtimes environments leveraging Python 3.5.5 and Python 3.9.

System Utilization:

Since this library depends on C++ with Python bindings, there will be close integration with the underlying system processes (memory, compute cores, architecture, file system, etc.).

Programming Language:

I intend to use Python 3.9 and C++. I will also use Docker containers as a means to test verify functionality on other operating systems.

Workload:

Since I am doing this project on my own, the target total time consumption is 20 hours. Breakdown:

- 4 hours familiarizing myself with the libraries, how to implement Python <-> C++ bindings, and obstacles I'm likely to encounter
- 3 hours writing test cases for Python 3.5.5 and Python 3.9
- 12 hours implementing library changes
- 1 hour documenting changes, creating a new tutorial, publishing to common repositories, etc.

- As time permits, I will extend my use case to include other library improvements and incorporate other Python 3.9 features. I would prefer not to introduce other library dependencies (e.g. CoreNLP, OpenNLP, etc.) unless there is a good reason and clear use case.