Andrew Rodriguez
10/17/12

HW 1 Report

I implemented a named entity recognizer (NER) that is heavily based on the starter code. Most of the work in putting together homework 1 involved working with the UIMA system: specifically developing my UML diagrams and mapping those ideas to the UIMA concepts and objects.

UIMA

I fit the NER process in the UIMA process: I created a CPE using the graphical tool that contains three steps: a collection reader (CollectionReader), an analysis engine (NERAnnotator), and a CASConsumer (Outputter). Each step performs what its name details: the collection reader reads hw1.in and produces a list of CAS's that represent each line. Each CAS contains the document text (the sentence) and a SourceId attribute that we add to the CAS's index which contains the document id. The next step in the pipeline is the NERAnnotator which analyzes the document text, runs the Stanford POS tagger on the text and performs my NER. Finally, the Outputter takes each CAS and for each detected NE and writes the correct output to the correct file. Outputter uses both the SourceId and NERAnnotation indices to get the relevant information for each sentence and print it.

General Dataflow

I implement perhaps the most simple UIMA process possible: there is only one Analysis Engine. The components are detailed in the above section.

NER construction

The NER process is heavily based on the started code. There are a couple of additions whose goal is to pick up a few of the lost NEs. The two implementations are an acronym detector and a set of “good word” detectors. In the guidelines for gene tagging it says that sequences that begin with a set of common words followed by gene words should be categorized as genes. I took this list of words and added them to a whitelist to avoid mislabelling some of these adjectives and adverbs. The acronym detector allowed all tokens that only consist of upper case characters.

NER evaluation

Most of the evaluation was ad-hoc examination of the results. I didn't want to bias my methods on the current input and output so I only considered a tiny bit at a time. More rules and evaluation would have had a great effect on the performance of the NER.