11-791 Design and Engineering of Intelligent Information Systems

Fall 2012 Assignment 1

Naoki Orii
norii@andrew.cmu.edu

October 17, 2012

Implementation of a Named Entity Recognizer with UIMA SDK

Figure 1 illustrates the overall data flow that occurs between the different types of components that make up the CPE.

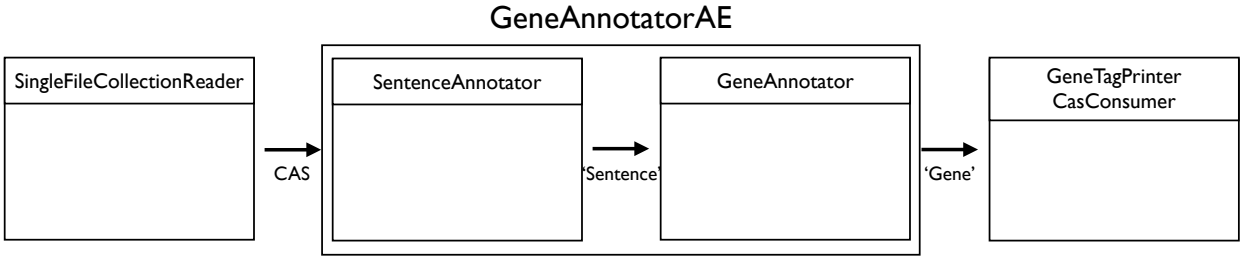

Figure 1: CPE components

The type system defines 2 types: `Sentence` and `Gene`. Both types have `sentenceID` and `rawString` as features. `sentenceID` stores the ID of the sentence containing the entity (e.g. "P00001606T0076"), while the `rawString` stores the raw string of the corresponding entity (i.e. for `Sentence`, it stores "Comparison with alkaline phosphatases and 5-nucleotidase", while for `Gene`, it stores "alkaline pho sphatases").

The implementation of the Collection Reader is `SingleFileCollectionReader`, which reads a single file and generates a CAS for each line in that file.

`GeneAnnotator` is an aggregate Analysis Engine made up from `SentenceAnnotator` and `GeneAnnotator`: `SentenceAnnotator` annotates `Sentence`s, and `GeneAnnotator` annotates `Gene`s. The main part of the gene annotation occurs in the `process` method of `GeneAnntoator`, which uses the `Chunker` class from LingPipe[1]. The gene mention tagging was split into 2 parts: (i) annotating sentences and (ii) annotating gene names (within a sentence), in order to decouple the gene annotation process from the data representation. In this way, if the format of the input file changes (i.e. if it came in xml format), we would not have to modify `GeneAnnotator`, but instead just need to write another annotator that generates `Sentence`s from a different input format.

The implementation of the CAS Consumer is `GeneTagPrinterCasConsumer`, which prints out `Gene` annotations in the required format.

[1] `http://alias-i.com/lingpipe/`