11-791: Design and Engineering of Intelligent Information System

Homework 2

Zexi Mao
zexim@andrew.cmu.edu

October 10, 2014

1 Architecture

1.1 Type system

The type system for the gene mention tagging task consists of the following types:

- `edu.cmu.deiis.types.Sentence`, which extends `org.apache.uima.jcas.tcas.Annotation`, is a type for input sentences. The text of a sentence is stored in the `org.apache.uima.jcas.JCas` object used to create the `Sentence` object. It also contains a feature `sentenceId`, which is a `org.apache.uima.cas.String` object for storing the unique ID for each sentence.
- `edu.cmu.deiis.types.Annotation`, which also extends `org.apache.uima.jcas.tcas.Annotation`, is a type for the recognized gene entities. Both the start-offset and end-offset are stored in the `Annotation` directly. An extra feature in `Annotation` is `entityText`, which is also a `uima.cas.String` object for storing the text piece recognized as a gene entity. The `casProcessorId` here is used to distinguish annotations made by different annotators.
- `edu.cmu.ediis.types.FinalAnnotation`. This type is quite similar to `Annotation` except that it is produced by the annotator that combines the single annotators.

1.2 Collection reader

The collection reader uses a `BufferedReader` to read the input file, each line (sentence) at a time and store the `sentenceId` and text into `Sentence`. The collection reader has a configuration parameter for the input file path, which is set to "hw2.in" in the release version.

1.3 Aggregate Analysis Engine

The aggregate analysis engine is composed of three annotators: `PosAnnotator`, `AbnerAnnonator`, and `CombineAnnotator`. `PosAnnotator` simply makes use of the part of speech tagger in Stanford Core NLP and simple named entity recognition rules to get the annotators. `AbnerAnnonator` uses a model trained on BioCreative corpus in ABNER. `CombineAnnotator` gets the annotations generated in the previous two annotators, checks whether each annotation generated by `AbnerAnnonator` has

overlap with annotations generated by `PosAnnotator`, if it does, create an annotation for the final result.

1.4 CAS consumer

The CAS consumer takes the produced `GeneEntity` and outputs all the annotated gene entity mentions to a file. It also has a configuration parameter for the output file path, which is set to "hw2-zexim.out".

2 Performance

Using the script provided, the performance of the system is shown as follows:

- Precision: 0.7395
- Recall: 0.6284
- F1-score: 0.6794