Part1. Designing and Implementing UIMA Analysis Engine

In this homework, I designed four annotators, along with their descriptors, composing of the aggregate analysis engine. According to the performance, I compared three methods of prediction,

1.1 Overview - Annotation

Since all annotations must share two common features, a super annotation class that describes the confidence and the annotator that generates this annotation (that is, casProcessorId) is important to lay a foundation for my system. In this system, the class of Annotation is designed for this purpose and all the other annotations can inherit from this super class Annotation to inherit these two features.

Other common features can also be added into Annotation if they are found later, and in this way, the consistence of system is promised. Begin and end position are also needed but both have already been built within the super class of org.apache.uima.jcas.tcas.Annotation.

1.2 Sentence Annotation

Basically, the system gets the input of a UIMA CAS, which is transformed from a text file given by the file path and name. Then the whole input is broken down to two levels: the sentence level and gene level. The sentence level gives its corresponding annotation, the SentenceAnnotation.

1.3 Gene Annotation

Except the sentence level, for the gene level, there are two types of annotations, Gene True Answer Annotation and Predicted Gene Annotation (including the Name Entity Annotation and the Abner Gene Annotation). Since they own the common features, Gene Annotation was designed as the parent annotation of the above two types of specific annotations.

1.3.1 Gene True Answer Annotation

In Gene True Answer Annotation, the gold standard answer was extracted from the provided file to generate "true answer" for our evaluable, which is a task-based standard to compare with the system output and give us the preliminary evaluation of the overall performance of the system.

1.3.2 Named Entity Annotation

In Named Entity Annotation, Stanford NER was applied to extract the annotation from the provided input file. Stanford NER is a Java implementation of a Named Entity Recognizer, which labels sequences of words in a text, such as gene and protein names. It comes with well-engineered feature extractors for Named Entity Recognition, and many options for defining feature extractors. NER was used as preprocess tool here for the actual application of Abner.

1.3.3 ABNER Gene Annotation

In order to improve the prediction performance, ABNER was applied in this homework. ABNER is a software tool for molecular biology text analysis. Its core is a statistical machine learning system using linear-chain conditional random fields (CRFs) with a variety of orthographic and contextual features.

1.4 Methods and evaluation

Three methods were applied and compared for their performance:

1. Only ABNER was applied:

Precision = 48%

Recall = 40%

F_measure = 44%

2. Used NER to run at first and applied the output from NER as input for ABNER

Precision = 65%

Recall = 28%

F_measure = 40%

3. Applied ABNER to get the annotation first and then compare the prediction with the output from NER, the intersection of both was extracted and the shorter annotations were selected as output.

Precision = 50.5%

Recall = 33.8%

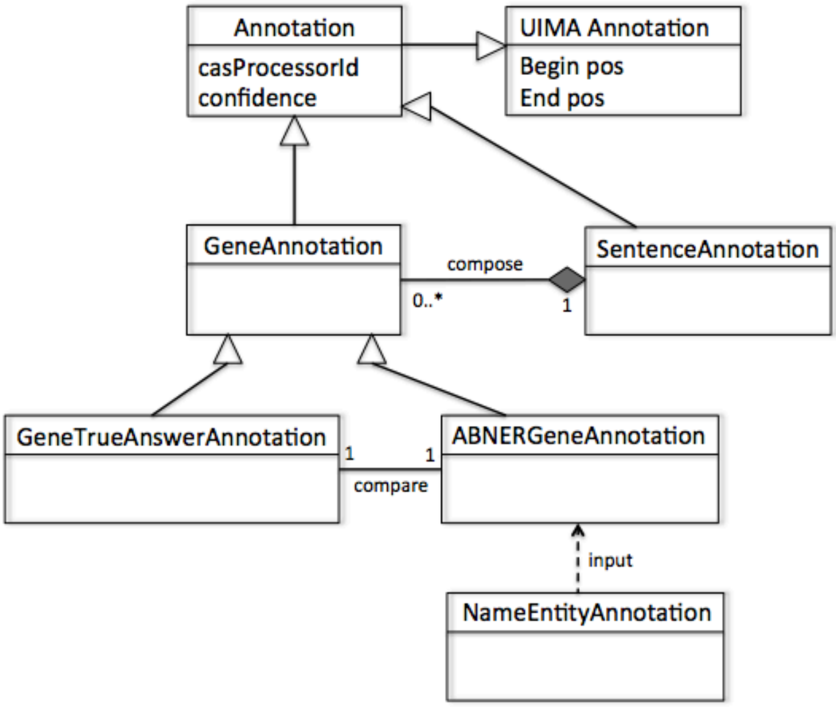
F_measure = 40.5%

Summary: According to my result, the second method retrieved the best performance.

Two things need to emphasize here:

1. To run my system appropriately, you need to add the Program argument < **./src/main/resources/CPEDescriptor.xml>** in Run Configurations - Arguments.
2. Javadoc was stored under: **/hw2-jli3/src/main/resources/docs/javadoc**

Part 2. Class Diagram of the system



Relationships included in this diagram are:

1. Inheritance: the relationship between Annotation and UIMA Annotation or between GeneAnnotation and Annotation are both examples of inheritance.
2. Association: the GeneTrueAnswerAnnotation and ABNERGeneAnnotation are in an association relationship
3. Composition: GeneAnnotation and SentenceAnnotation are in composition relationship.
4. Input: NameEntityAnnotation and ABNERGeneAnnotation are in input relationship.