IIS — hw1

Chu–Cheng Lin `chuchenl`

September 2014

1 UML Design

See Fig. 1 and Fig. 2.

2 System Description

At its core, the system uses a statistical named entity recognizer provided by Lingpipe. The recognizer is HMM based and pre–trained.

The system is rather simple. There is a collection reader called `MyCollectionReader`, an annotator called `GeneAnnotator`, and a consumer called `MyConsumer`. There is also a wrapper class for the Lingpipe package we used, called `Chunker`. Following is the description of the components.

`MyCollectionReader` reads the input, which is specified in the descriptor by the parameter name `InputFileName`. The specified file is opened and read in `initialize`. `MyCollectionReader#getNext` gets the content of the next line in the file through an iterator. We pass on the line as a `Sentence` annotation, which has two self–explanatory properties `id` and `text`.

`GeneAnnotator` has an `initialize` method where it initializes the wrapper `Chunker c` with the model file path specified in the configuration parameter `ModelPath`. The constructor of `Chunker` in turn loads the specified model file using the API provided by Lingpipe. The `process` method retrieves the `Sentence`s of the input annotation. The `text` property of every `Sentence` is fed into the named entity recognizer through the wrapper metho `Chunker#chunk`, which returns a `Set` of chunked named entites. For each recognized named entity, `GeneAnnotator` creates a new annotation `GeneAnnotation`, which contains the properties `id`. We use the inherited properties `begin` and `end` to store offsets, which we also compute in `GeneAnnotator`.

`MyConsumer` gets the specified output filename from the configuration parameter `OutputPath` in the method `initialize`. It retrieves the `GeneAnnotation`s associated with an annotation, and and writes into the output file in the specified format.
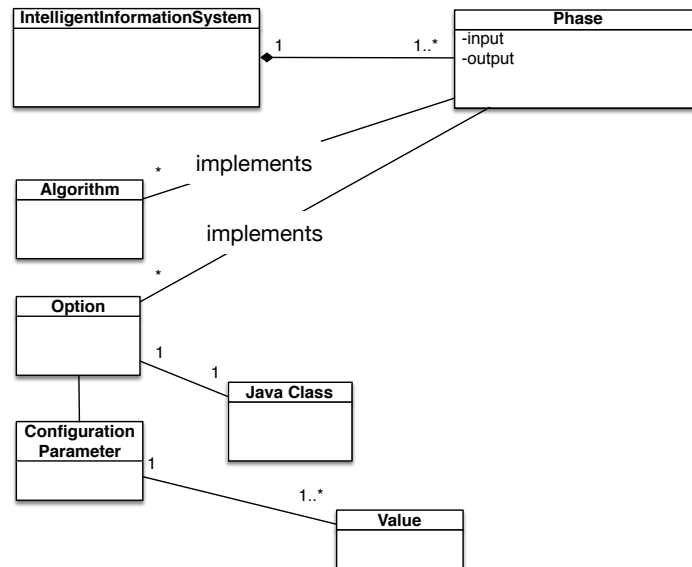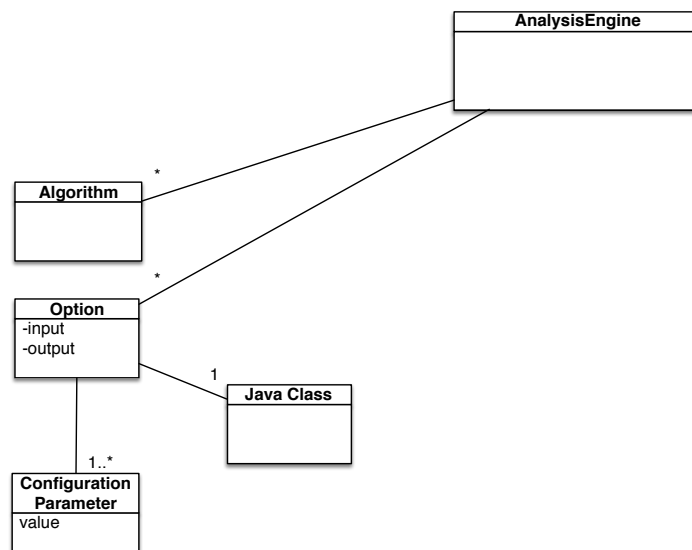
Figure 1

IntelligentInformationSystem

Phase
-input
-output

1 ... 1..*

Algorithm

* implements

Option

* implements

1 ... 1

Java Class

Configuration Parameter

1 ... 1..*

Value

Figure 1: Task 1.1

Figure 2

AnalysisEngine

Algorithm

*

Option
-input
-output

*

1

Java Class

1..*

Configuration Parameter
value

Figure 2: Task 1.2