HW-1 REPORT

Name: Prajwal Yadapadithaya
AndrewID: pyadapad

Architecture:

This Gene Named Entity Recognizer is implemented using the UIMA framework, with making use of the LingPipe toolkit for text processing. In this section, I will be describing the implementation of the type system and other descriptors used in the UIMA pipeline for Gene Named Entity Recognition.

Type System: *GeneTypeSystemDescriptor.xml*
The type system describes the feature names used in the pipeline. The different types used in this system are as follows:
project.deiis.types.GeneAnnotator - has *casProcessorId*
project.deiis.types.InputData - Consists of *sentenceId* and *geneData*
project.deiis.types.Results - Consists of *senetenceId, geneProduct, geneStartOffset* and *geneEndOffset*

Collection Reader: *GeneCollectionReader.xml (Implementation: GeneCollectionReader.java)*
The collection reader is responsible for reading input text line by line, and splitting the sentenceId and the string containing gene information *(geneData)* and passing the information to the analysis engine using the *InputData* feature in the Cas.

Analysis Engine: *GeneAnalysisDescriptor.xml*
Analysis engine comprises of a single annotator in this implementation. This is responsible for triggering the execution of GeneDataProcessor.

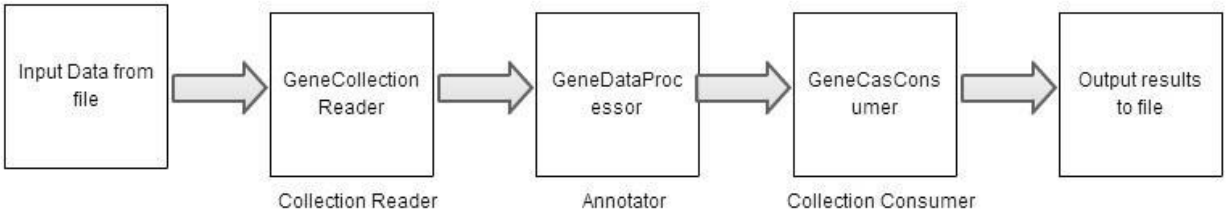
Annotator: *GeneDataProcessor.xml (Implementation: GeneDataProcessor.java)*
GeneDataProcessor is responsible for using the LingPipe named entity recognition library to extract the gene names based on a trained model (GENTAG). *GeneDataProcessor* is also responsible for calculating the start and end offsets for the output, and passing the output to Cas Consumer using the *Results* feature.

Cas Consumer: *GeneCasConsumer.xml (Implementation: GeneCasConsumer.java)*
GeneCasConsumer is responsible for writing the results to the specified output file. *GeneDataProcessor* sends an entry for each gene name obtained to the consumer, which processes them and writes them to the output file.

Algorithms:

The Analysis Engine in this implementation has an annotator called *GeneDataProcessor* which uses LingPipe library for processing the input text and extract gene data. The LingPipe named entity recognizer expects a string to be given as input on which it does the required processing to extract gene names. The LingPipe Chunker class is trained using the model file (*gene_model*), and it returns slices of the input string which match to the gene names obtained from the training model. Internally, LingPipe provides an implementation for the Aho-Corasick string matching algorithm, which is used for named entity recognition. I have included the model file used for this implementation in the project.

Data Flow in the System:



The data flow follows a typical UIMA architectural data flow. As described in the architecture, in this implementation, *GeneCollectionReader* (Collection Reader) reads from the input file line by line, splits the sentence ID and text in which we are interested, before passing the information to *GeneDataProcessor* (Annotator) which uses LingPipe library to check the data and extract gene information with the help of an existing training model. It then generates the required data for the *GeneCasConsumer* (Consumer), which finally writes the extracted gene information to the output file (hw1-pyadapad.out)

Experiments:

The pipeline was tested with the input file which was given in the archetype. The output generated was compared with the sample output file given in the archetype.

Results:

This gene named entity recognizer implemented using UIMA and LingPipe toolkit is able to extract gene names out of the given input file in less than 8 seconds. The output generated has all the gene names mentioned in the sample output file. However, there are some extra names extracted which are not present in the sample output file. This could be because the model on which the named entity recognizer is working is not perfect. An improvement would be to implement another algorithm for named entity recognition and make a decision based on the results from both the algorithms, using a confidence attribute.

Reference:

I have referred the following links for implementing gene named entity recognizer using UIMA framework.

- http://uima.apache.org/downloads/releaseDocs/2.1.0-incubating/docs/html/tutorials_and_users_guides/tutorials_and_users_guides.html
- http://alias-i.com/lingpipe/demos/tutorial/ne/read-me.html
- https://svn.apache.org/repos/asf/uima/uimaj/trunk/uimaj-tools/src/main/java/org/apache/uima/tools/components/FileSystemCollectionReader.java (Example Collection Reader)