Name: Venkata Puneet Ravuri
Andrew ID: vravuri

Design and Implementation Report

1. Abstract:

This project deals with annotating gene names from an input dataset using named entity recognizers and rule based grammars. The input data will be a document containing multiple lines of biology specific data, each line identified by a unique formatted code. The goal of the project is to identify these genes for every line and present an output of all the identified annotations.

2. Design:

The UIMA interface provides necessary framework for the design. There are three major components in the design – Input parsing, analyzing the input to create annotations and finally printing the annotations to an output source. These three tasks are to be performed serially in the correct order.

Input Parsing:

Input parsing has been implemented using the Collection Reader framework of UIMA. One collection reader has been implemented that takes input file as a parameter and parses the document. It then uploads the document as a string to the Common Analysis System (CAS).

Gene Name Annotations:

Creating annotations is a complex process that has to be done in multiple phases since the document has to be analyzed in multiple perspectives. UIMA supports an Aggregate Analysis Engine for this purpose which has been used for this project. The process of annotation has been divided into three phases:-

- Identifying annotations based on certain keywords which frequently occur in protein names.
- Identifying annotations based on abbreviation patterns occurring in sentences.
- Identifying annotations related to protein p53 specific gene names since they follow specific grammatical rules.

For each of these phases, a separate analysis engine has been implemented. These engines are independent of each other and add the respective annotations to the CAS object. The aggregate

analysis engine holds references to these three "primitive" analysis engines. To control the flow of calling the analysis engines, UIMA provided Flow Controller interface has been used. A flow controller has been implemented that calls the analysis engines one after the other (since there is no specific dependency between each of the analysis engines). The aggregate analysis engine refers to the flow controller to call the individual analysis engines.

The individual analysis engines perform the annotations. The Stanford NLP has been used to identify named entities within the strings. The frequently occurring keywords like "protein", "vasopressin", etc will be used along with named entities to identify the gene names. Generally, gene names are camel-cased and some of them are alphanumeric words. These properties are also taken into consideration when recognizing the gene tags.

Validation and Output:

The third and final component is the validation and output of the annotations. For this purpose, UIMA provided CAS Consumer interface is used. The CAS consumer in this project collects all the annotations, removes duplicates, sorts them based on the unique line code of the sentences and then prints them into the file that has been taken as a parameter.

Data Types:

For every annotation, there are four parts – unique code of the line from where the annotation is present, the beign offset of the annotation, the end offset of the annotation and the actual annotation string. To store these artifacts for every annotation, one datatype has been created in the type system. The same type system has been used by all the analysis engines since all the annotators deal with same type of annotations.