Final Report

Andrew ID: leixiao

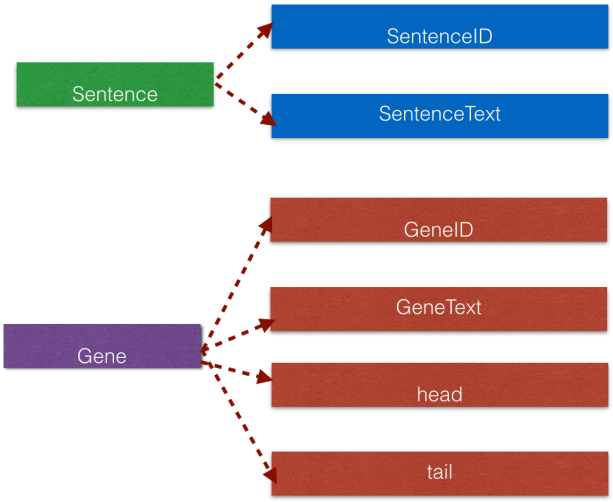
Name: Lei Xiao

1. Architecture Design

(1) Type System

In this part, I designed two types in it, which include Sentence and Gene.

Sentence has two functions, namely SentenceID and SentenceText. This type is used to deal with the mission that when reading the input file, the CollectionReader can automatically turned the input file into several sentences. And then separate the sentence into SentenceID, which could be used later for identifying the GeneID, and SentenceText, which part would be used to analyze the gene tags. The partition method is to identify the location of the space(“ ”), and the part from the beginning of the sentence to the space can be recognized as SentenceID, and the rest part is SentenceText.

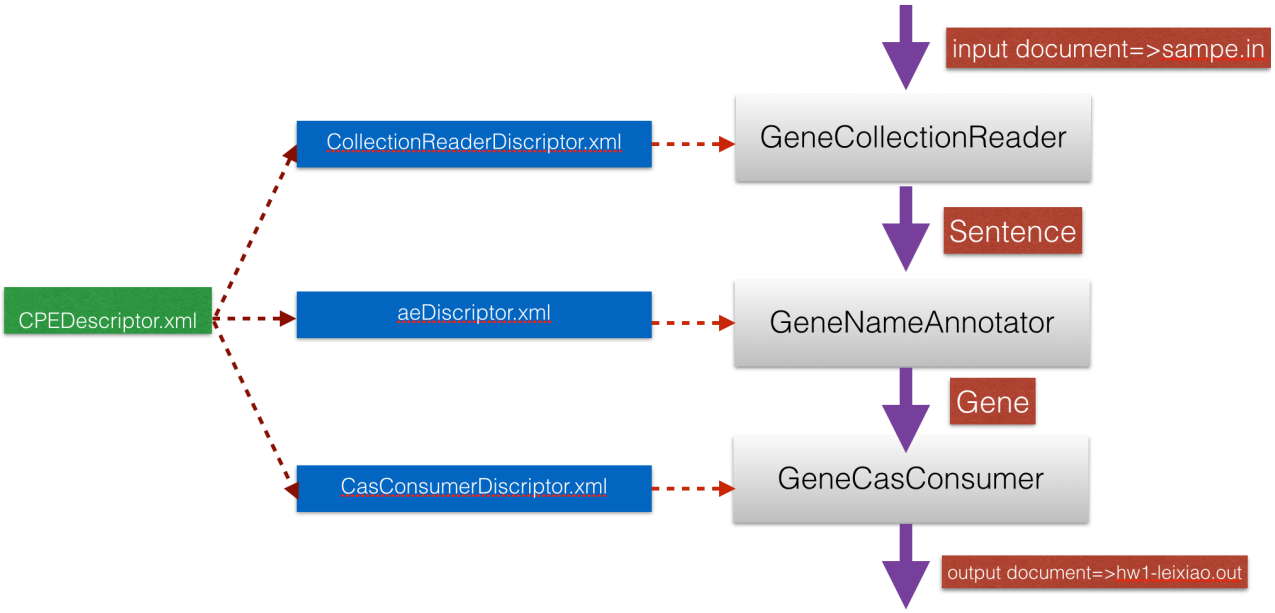


Gene has four functions, namely GeneID, GeneTag, head, and tail. This type is used in the annotator and casConsumer part. Gene is the analyzing result of the Sentence type, and we can say that the annotator changes the Sentence type into gene type, and then sends it to the casConsumer, which could put the content of CAS in the output document. GeneID is the same as the SentenceID, which is used to find which sentence the gene tag belongs to. GeneTag is the analysis result from the annotator, which we get from the name entity recognizer. Head and tail means the begin position and end position of each gene tag of a specific sentence, which satisfies the output format. And head and tail are the return results (as a format of map)of the entity recognizer.

(2) Engineering Process

In this part, I will introduce the whole architecture of the process.

The following picture shows the process of dealing with data and specific types in each component of the system.



(3) Design Pattern

As can be seen from the above picture, the whole design of the system can be divided into three parts, namely GeneCollectionReader, GeneNameAnnotator and GeneCasConsumer. Each component has its own functions and roles in this system.

1) GeneCollectionReader, whose function is to read from the input document, for example, sample.in, and then separate the document into several sentences, and then give the value to SentenceID and SentenceText. The main idea of this part is to get the initial format (Sentence) from the input file and transfer it to the annotator to process it later.

2)GeneNameAnnotator, whose function is to turn the input Sentence into the Gene type we want to get. The main idea of this part is to deal with the SentenceText and recognize the location of the gene tag. Actually, it is really a hard task, so what I do is just to use the name entity recognizer the instructors give. Hence, I can only get the noun, but not the nouns which is related to gene area, Maybe add some algorithms here will be useful, for example, we could add a gene database here, which includes enough

gene related nouns. So, when we get the nouns, we should turn to check if the noun exists in the database, if is, we can store it in the GeneTag, but if not, we should ignore it.

3)GeneCasConsumer, whose function is to get the gene type and then output them in the document, in this case, we put it into the hw1-leixiao.out. The main idea of this part is to obtain the gene type, transfer it into the format we want to store in the document, and then put it into the outputfile.

(4) Additional Design

The system need several descriptors to help design the functions of each component. For example, we should build ColletionReaderDiscriptor.xml to describe the parameter, type system, capabilities, and indexes of the GeneCollectionReader.

In addition, we need to build a CPEDescriptor to implement all the components together to deal with the whole data process, which could be recognized the controller of the whole process. The method of building CPEDescriptor is from the UIMA CPE GUI, which could automatically build the CPEDescriptor.xml.

2. Algorithm Design

1)Divide the sentence into id and text

To identify the location of the space(" "), and the part from the beginning of the sentence to the space can be recognized as SentenceID, and the rest part is SentenceText.

2)Use Name Entity Recognizer from Stanford

Put the SentenceText into this method, it will automatically return a Hashmap. Through traversing the hash map, we could get the begin and end of each gene tag of the sentence .

3. Additional information

I read the source code of the DEIIS system and learn the whole process of dealing UIMA related framework. And then I read related materials of UIMA tutorial and run some examples of the uimaj-examples to understand the details of the three main components of the system, namely CollectionReader, NameAnnotator and CasConsumer.