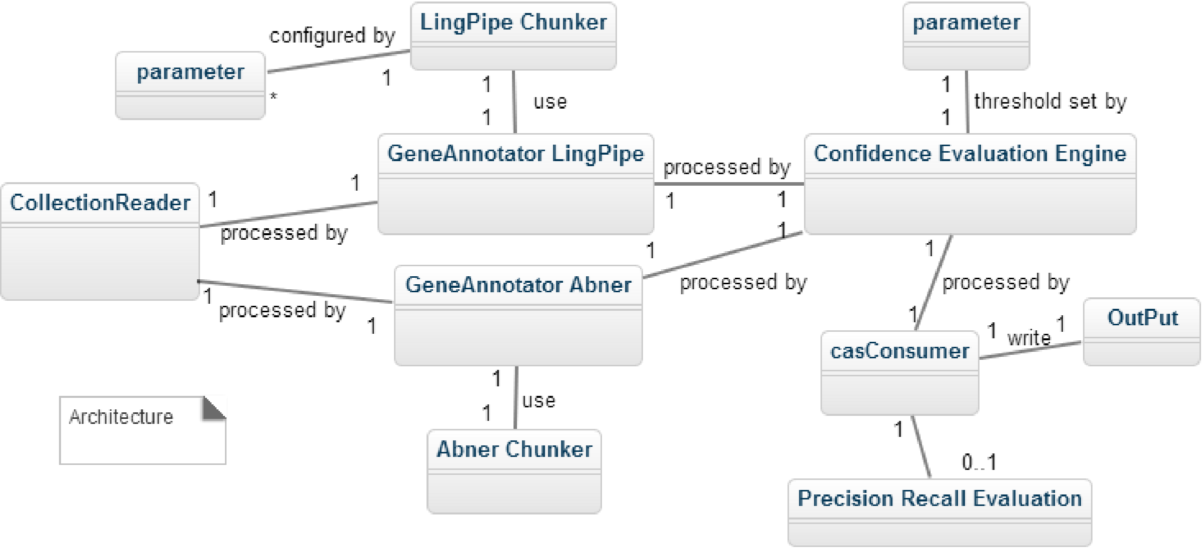
Name: Zhoucheng Li

Andrew ID: zhouchel

Homework 2

1 Architecture Overview

UML



This architecture mainly contains three parts (excluding the input documents), i.e. Collection Reader, Aggregate Analysis Engine and casConsumer.

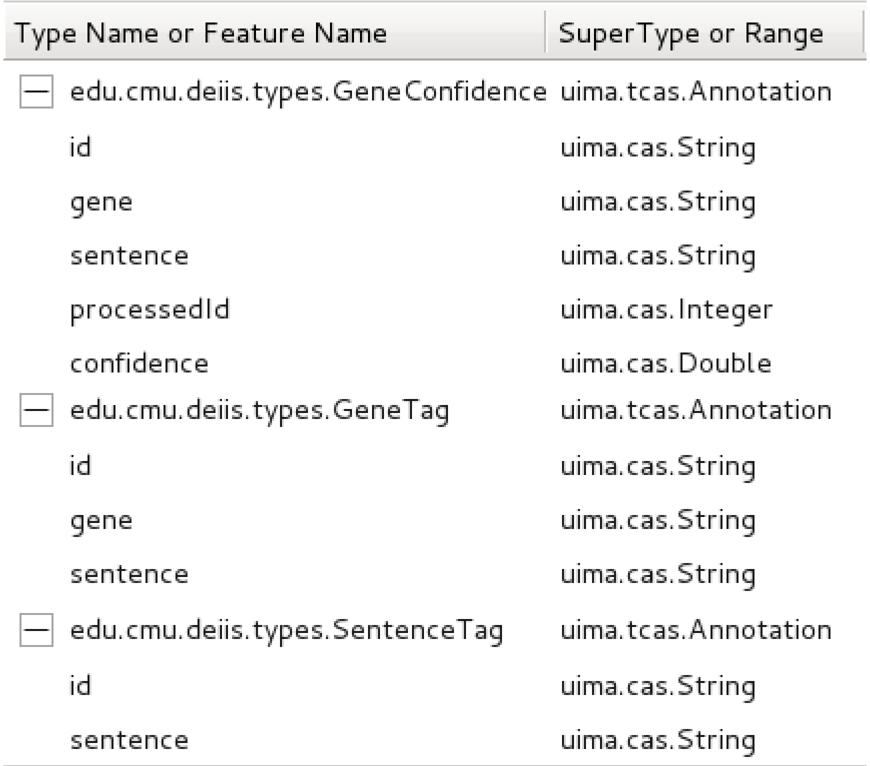
The Collection Reader would read data line-by-line from the document, separate the each line into ID and text, and wrap them into CAS indices.

The Analysis Engine is an aggregate analysis engine which contains three primitive analysis engines, namely GeneAnnotator using LingPipe, GeneAnnotator using Abner and a Confidence Evaluation Engine which combines results from previous annotators.

Finally, casConsumer would iterate all GeneType Feature Structures, revise the positions by calculating whitespace-excluded offsets, and write all results into a file. The CasConsumer can also do an evaluation process if evaluation option is specified in a configuration file and gold standard output file parameter is set in the casConsumer.xml.

2 Detailed Implementation

2.1 Type System



Type Name or Feature Name	SuperType or Range
⊟ edu.cmu.deiis.types.GeneConfidence	uima.tcas.Annotation
id	uima.cas.String
gene	uima.cas.String
sentence	uima.cas.String
processedId	uima.cas.Integer
confidence	uima.cas.Double
⊟ edu.cmu.deiis.types.GeneTag	uima.tcas.Annotation
id	uima.cas.String
gene	uima.cas.String
sentence	uima.cas.String
⊟ edu.cmu.deiis.types.SentenceTag	uima.tcas.Annotation
id	uima.cas.String
sentence	uima.cas.String

There are there types (***SentenceTag***, ***GeneConfidence*** and ***GeneTag***) in this system. They are all inherited from ***uima.tcas.Annotation*** which defines two integer features indicating begin and end of the span being annotated.

SentenceTag is used for input which annotates sentences and corresponding IDs.

GeneConfidence is the temporary type that holds the results of two gene annotators (i.e. with LingPipe and Abner). This type has a “processedId” feature which is used to discriminate the results of LingPipe from Abner. “confidence” feature is used to measure the probability of a gene name being true positive.

GeneTag is used for output which annotates gene name’ positions and original sentences and corresponding IDs.

2.2 Collection Reader

Several functions (arguments are omitted here) like initialize(), next(), hasNext() and close() are rewritten in Collection Reader.

In Initialize(), document would be opened. hasNext() is used to check if there’s next line in the document. If there exists next line, we can use next() to process this line, separating it into ID and text, wrapping ID and text into a SentenceTag, adding the annotation into CAS index. When this document is processed over, close() would close the document.

2.3 Analysis Engine

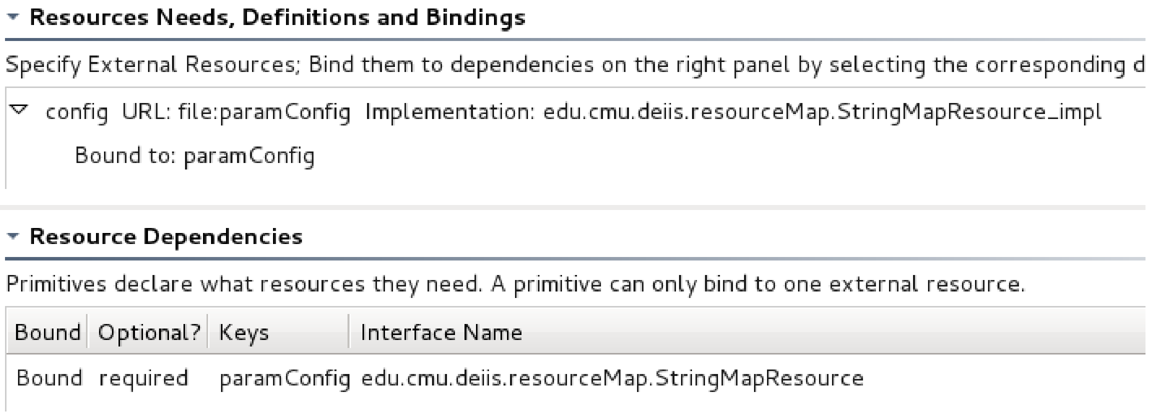
2.3.1 Configuration File

Several parameters are specified in *paramConfig* using following format:

```
MAX_N_BEST_CHUNKS = 30
LingPipeThreshold = 0.1
N-Best_NER = true
ConfidenceThreshold = 0.7
Evaluation = false
```

The left part of "=" servers as key and the right part of "=" is treated as associated value to that key.

UIMA provides an easy way to access external resources from UimaContext.



Resources Needs, Definitions and Bindings

Specify External Resources; Bind them to dependencies on the right panel by selecting the corresponding d

config URL: file:paramConfig Implementation: edu.cmu.deiis.resourceMap.StringMapResource_impl

Bound to: paramConfig

Resource Dependencies

Primitives declare what resources they need. A primitive can only bind to one external resource.

Bound	Optional?	Keys	Interface Name
Bound	required	paramConfig	edu.cmu.deiis.resourceMap.StringMapResource

Here, StringMapResource is an interface which separates the resource file from annotators (i.e. StringMapResource_impl would directly access the configuration file while annotator cannot).

In class StringMapResource_impl, configurations are read and separated into "key-value" pairs and stored in a public *static* HashMap. So other components like casConsumer could also access to the configurations latter.

2.3.2 Gene Annotator using LingPipe

This annotator uses LingPipe toolkit to annotate all gene mentions.

In class GeneAnnotatorWithLingPipe, I rewrote the initialize() function so I could load the configuration file before process phrase. Also I could initialize the NER before process phrase. N-Best algorithm and Confidence chunker are adopted in this homework. So the confidence of the gene name could be calculated.

MAX_N_BEST_CHUNKS are set to 30 and the threshold for LingPipe is set to 0.1 so that it could have much higher recall and filter those results that have very low confidence.

2.3.3 Gene Annotator using Abner

Abner's core is a machine learning algorithm based on CRFs. It provides a gene tag model trained on the **BioCreateive** Corpora.

Abner toolkit doesn't provide confidence information about the extracted gene name. So I give it full-score (1.0) if it is classified as gene name by Abner.

One interesting thing is that Abner treat many gene names as "Protein", for example, "alkaline phosphatases". So I mark all Proteins as Gene names when using Abner.

When use Abner stand-alone, the precision and recall could reach:

```
Precision: 0.781535158347
Recall: 0.637722419929
F1 Score: 0.702342548765
```

which is very close to the tested precision and recall posted on its website. So compared to LingPipe's stand-alone test result, Abner's test result is more reliable because it trained the model in a corpora different from the sentences from hw2.in.

2.3.4 Confidence Evaluation Engine

This annotator would fetch all ***GeneConfidence*** , and build a HashMap to store all gene's confidence.

1. If a gene name only appears in LingPipe or Abner, use their original confidence as final confidence.
2. If a gene name could be found by both LingPipe and Abner, this gene name is much likely to be a correct one, so I use the sum of LingPipe and Abner's confidence as final confidence.

After combining all confidences, this evaluation engine would judge whether or not a gene name should be sent to casConsumer based on its final confidence score. If the confidence is larger than the pre-set confidence threshold, add this gene to ***GeneTag*** and update the CAS.

CasConsumer

2.3.5 Position Revision and Write Output File

In CasConsumer, Gene annotations would be iterated using Feature Structure iterator. Begin and end positions were updated with calculated whitespace-excluded offsets. Sentence ID, gene name, begin and end positions were written into output file. The file path is specified in casConsumer.xml.

2.3.6 Evaluation Process (optional)

In case there is no specified golden standard output file. I use an option "Evaluation = true/false" in configuration file to decide whether perform evaluation or not. Also, if evaluation option is set to "true", stand gold output file should be placed at the root of the project and be named as "hw2.gold_stand".

The evaluation function is really simple. First, set a variable *TruePositive* to 0, read each line in golden standard output file, add it a HashSet. Then read each line in predicted output, check if this line is contained in previously generated HashSet. If it's in there, add one to *TruePositive*, else continue. Then the precision and recall could be calculated as:

$$\text{Precision} = \frac{TP}{N_{\text{Predicted}}}$$

$$\text{Recall} = \frac{TP}{N_{\text{GoldStandard}}}$$

3 Algorithms and Model Resources

3.1 Baseline

The baseline algorithm uses Stanford coreNLP library. It will extract all nuns and thus the precision of base line is very low. This part is done in HW-1.

```
Precision: 0.1025
Recall: 0.5467
F1 Score: 0.1727
```

Adding Postagger in HW-1 wouldn't help improve the F1 score, so in this homework, it is discarded.

3.2 LingPipe NER with Statistical Model

The model has been trained from GENETAG corpus. Download:HmmChunker Model

The selection of algorithm (First-Best, N-Best or Confidence N-Best) depends of the configuration file.

3.3 Abner

It contains two models trained on the **NLPBA** and **BioCreative** corpora. The **BioCreative model** has one entity (subsuming genes and gene products) trained on 7,500 sentences, evaluated on 2,500.

4 Experiments

Using the experiments result in HW-1, I choose N = 30 for LingPipe's Confidence N-Best-Chunker and very low threshold 0.1 (actually 0.0 is also viable) which could improve the whole recall. However, since LingPipe's model actually trains on partial of the test data for HW2, so the stand-alone in HW1 experiments are convincible. That's why I give Abner much higher score (full-score) if it finds a gene. For the same reason, the final confidence threshold is set to be low than 1.0 so that all Abner's results will be treat as believable. And gene names only appear LingPipe could be regarded as gene if and only if it's score is higher than the final confidence threshold.

The final result for HW-2:

```
Precision: 0.744581899358
Recall: 0.876539830276
F1 Score: 0.805190233108
```