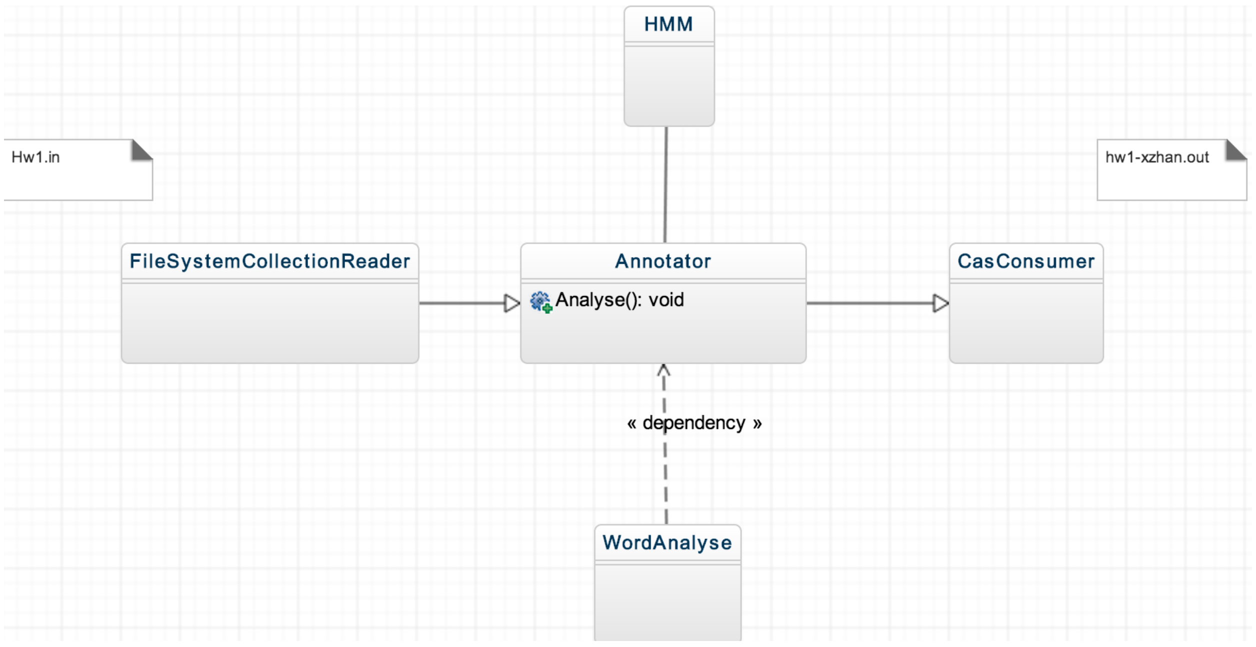
ID: xzhan HW1 Xiaoran Zhan

Design Pattern

Since this system is quite obvious, every class has its own responsibility, so I build it with Information Expert Pattern.

Rough dataflow



The general idea of my system is:

1. Build a FileSystemReaderCollection, read the sample data file, and put the whole document and location of file into CAS.

This part I directly used the file from UIMA example. Since I don't have to use the doc location so I delete it.

2. After storing all the doc content, build the annotator to analyze the the content of doc.

I firstly used the stanford NLP, but it has bad performance, so I turn to the GENETAG HMM for Genetag of Lingpipe, which is a well trained model for gene recognition. Since it's already trained, I can directly use it to analyze the content. The code to invoke theme are quite similar.

Divide the doc by line, then input each line into GENETAG HMM to analyze. In a for loop, construct the annotation of my type, WordAnalyser. Since it's already trained, I don't have to include gene tag when designing the type system. The wordAnalyser only consist two attribute, the id and sentence. Id is used to store line number, sentence is used to store content of each line.

Construct the annotation for each word the GENETAG HMM thinks is gene, store the start and end index of word. Since the index of word in sample.out doesn't take the space into account, so we have to modify the index of word a little bit. Also put id, sentence of the word into annotation, add the annotation to index.

3. Use the CasComsumer to output our analyzing result.
Use the jcas to get index of annotations constructed before, iterate through all the result and print them in the same form as the sample.out.

So the whole process is over.

Performance experiment

I also build a test class in src/test/java/Test/AccTest.java to test the performance of result from both StanfordNLP and HMM.

The result from StanfordNLP:

```
Hitting number: 9922
Number of sample.out: 18265
Number of hw1-xzhan: 97375
The precise of our system is 0.10189473684210526
The recall of our system is 0.5432247467834657
```

Since it only extract none, it is expecting that such low result will appear. So I decisively turned to HMM.

The result from HMM:

```
Hitting number: 15420
Number of sample.out: 18265
Number of hw1-xzhan: 20174
The precise of our system is 0.7643501536631307
The recall of our system is 0.844237612920887
```

The precise and recall improve dramatically. Admitted that is still much room of promotion in terms of precise.

About HMM
I look look into the result of out and found many unmatched result like:
“S-N-Pog” Actually should be a position.
“Retinoic acid-related Orphan receptor Response Element)-like sequences”
“protein-protein”

Seems like it is more likely to include form like “B-C” thus make it precise much lower.

I tried to find any doc about GENETAG HMM but failed.

Question

I do have a question.
When Constructing annotator, I instantiate each annotation in double loop, as a result it cause out of memory exception. I allocate more memory for my system. However I know this is not the best way to solve it. So could you give some advice?

Thank you!