Report for Execution Architecture with CPE and Deployment Architecture with UIMA-AS

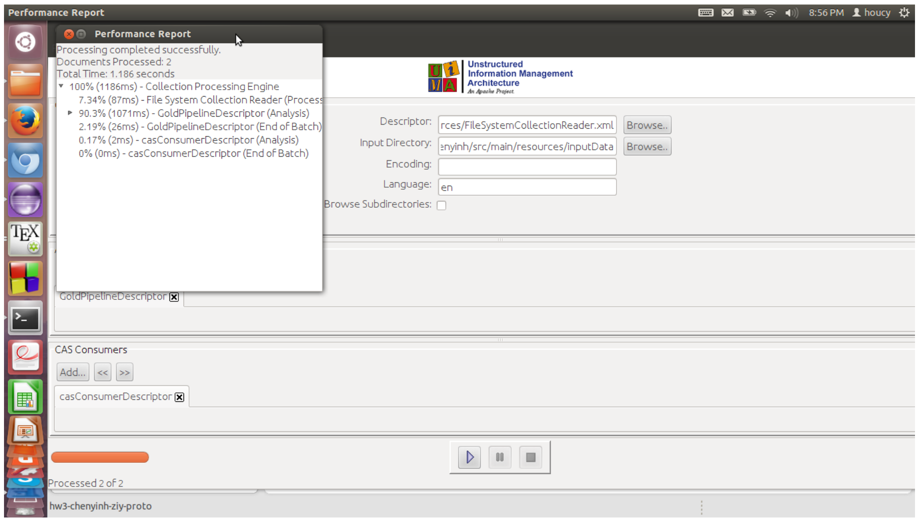
Chenying Hou
Andrew ID:chenyinh

October 7, 2013

1 Introduction

This homework has two main parts. The first part is to build CPE with Collection Reader, hw2 aae and casConsumer. The second part is about deployment architecture with UIMA-AS. The main difficulties I met in this homework are understanding of CPE(the concept of Collection Reader and casConsumer), UIMA-AS mechanism (such as the concept of broker, and the difference between UIMA-AS aggregate and UIMA aggregate) and how to integrate remote service into my own analysis engine.

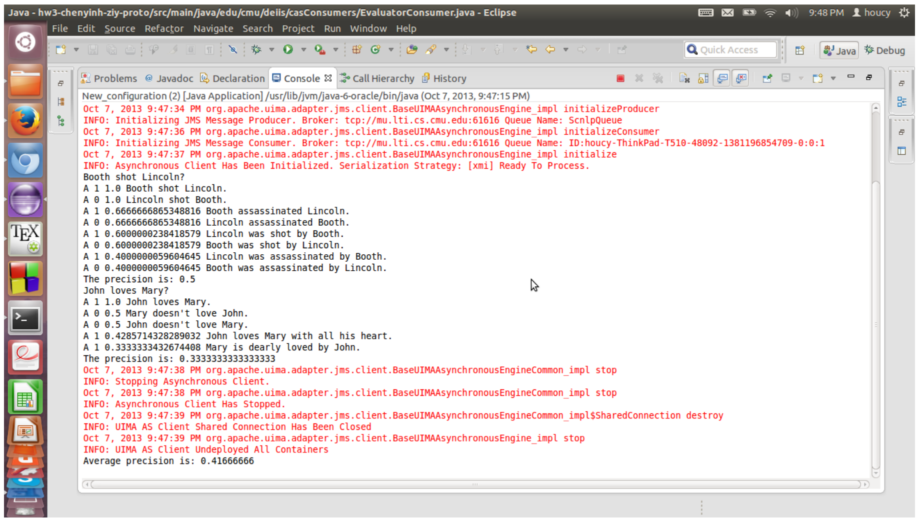
2 Execution Architecture with CPE


```
Processing completed successfully.
Documents Processed: 2
Total Time: 1.186 seconds
▾ 100% (1186ms) - Collection Processing Engine
    7.34% (87ms) - File System Collection Reader (Process
  ▸ 90.3% (1071ms) - GoldPipelineDescriptor (Analysis)
    2.19% (26ms) - GoldPipelineDescriptor (End of Batch)
    0.17% (2ms) - casConsumerDescriptor (Analysis)
    0% (0ms) - casConsumerDescriptor (End of Batch)
```

Descriptor: rces/FileSystemCollectionReader.xml
Input Directory: enyinh/src/main/resources/inputData
Encoding:
Language: en
Browse Subdirectories: ☐

Processed 2 of 2

Figure 1: CPE for hw2


Java - hw3-chenyinh-ziy-proto/src/main/java/edu/cmu/deiis/casConsumers/EvaluatorConsumer.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Problems
Javadoc
Declaration
Console
Call Hierarchy
History

```
New_configuration (2) [Java Application] /usr/lib/jvm/java-6-oracle/bin/java (Oct 7, 2013, 9:47:15 PM)
Oct 7, 2013 9:47:34 PM org.apache.uima.adapter.jms.client.BaseUIMAAsynchronousEngine_impl initializeProducer
INFO: Initializing JMS Message Producer. Broker: tcp://mu.lti.cs.cmu.edu:61616 Queue Name: ScnlpQueue
Oct 7, 2013 9:47:36 PM org.apache.uima.adapter.jms.client.BaseUIMAAsynchronousEngine_impl initializeConsumer
INFO: Initializing JMS Message Consumer. Broker: tcp://mu.lti.cs.cmu.edu:61616 Queue Name: ID:houcy-ThinkPad-T510-48092-1381196854709-0:0:1
Oct 7, 2013 9:47:37 PM org.apache.uima.adapter.jms.client.BaseUIMAAsynchronousEngine_impl initialize
INFO: Asynchronous Client Has Been Initialized. Serialization Strategy: [xmi] Ready To Process.
Booth shot Lincoln?
A 1 1.0 Booth shot Lincoln.
A 0 1.0 Lincoln shot Booth.
A 1 0.6666666865348816 Booth assassinated Lincoln.
A 0 0.6666666865348816 Lincoln assassinated Booth.
A 1 0.6000000238418579 Lincoln was shot by Booth.
A 0 0.6000000238418579 Booth was shot by Lincoln.
A 1 0.4000000059604645 Lincoln was assassinated by Booth.
A 0 0.4000000059604645 Booth was assassinated by Lincoln.
The precision is: 0.5
John loves Mary?
A 1 1.0 John loves Mary.
A 0 0.5 Mary doesn't love John.
A 0 0.5 John doesn't love Mary.
A 1 0.4285714328289032 John loves Mary with all his heart.
A 1 0.3333333432674408 Mary is dearly loved by John.
The precision is: 0.3333333333333333
Oct 7, 2013 9:47:38 PM org.apache.uima.adapter.jms.client.BaseUIMAAsynchronousEngineCommon_impl stop
INFO: Stopping Asynchronous Client.
Oct 7, 2013 9:47:38 PM org.apache.uima.adapter.jms.client.BaseUIMAAsynchronousEngineCommon_impl stop
INFO: Asynchronous Client Has Stopped.
Oct 7, 2013 9:47:39 PM org.apache.uima.adapter.jms.client.BaseUIMAAsynchronousEngineCommon_impl$SharedConnection destroy
INFO: UIMA AS Client Shared Connection Has Been Closed
Oct 7, 2013 9:47:39 PM org.apache.uima.adapter.jms.client.BaseUIMAAsynchronousEngine_impl stop
INFO: UIMA AS Client Undeployed All Containers
Average precision is: 0.41666666
```

Figure 2: Average Precision Score of Overlap Scoring Method

Figure 1 shows my CPE built for hw2 aae. Using FileSystemCollectionReader, I need to build a .xml file for it. Different from hw2, we do not use Document Analyzer and instead of that, I integrated the whole process together using CPE. CPE can divide and integrate the whole processing task elegantly. Collection Reader part is responsible for all input reading task and generates JCas for the analysis engines. Then the aae developed for hw2 is the middle processing part and conducts annotating jobs. In the last part, casConsumers can use the data in JCas and process data as we want such as the evaluation part.

The idea of splitting this process into these parts is really cool. In the hw2 I felt a little bit confusing that I wrote evaluation as a annotator but it does not add any new things into JCas. In this homework, I feel it is very reasonable to put evaluation part in casConsumer. And actually casConsumer can do more than evaluation jobs such as analysing more data feature based on the JCas generated by aee.

2.1 Calculating average precision

In this task, after learning CPE, I also solved one problem in homework 2 that how to compute the average precision. I learned that *collectionProcessComplete()* method can be executed at the end of the collection, so I used two variable to record the accumulate precision and number of questions and then computed the final average precision and outputted it in this function. Figure 3 is the result of the average precision using NameEntity in scnlp.

3 DeploymentArchitecture with UIMA-AS

In this task, I spent time learning what is UIMA-AS and how to call remote service and how to deploy my own UIMA-AS service. In this part, I began to realize the strong power of UIMA-AS.

3.1 Call remote service

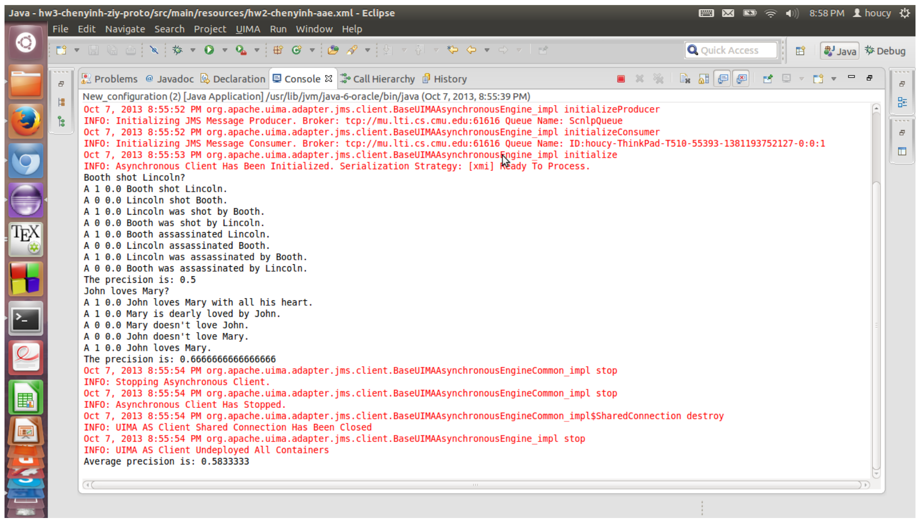
Calling Stanford CoreNLP service is not hard since it just need to build a scnlp client xml file. And Building the scnlp client file is easy especially using GUI (adding remote analysis engine).

3.2 Computing AnswerScore using NameEntity Mention

I also considered how to integrate scnlp service into my hw2 aae and compute a new score by using scnlp NameEntity. I found that NameEntity Mention can record type information for each token, such as whether it is *person*, *number* or null. So, I used this Mention type information to calculate the overlap between questions and answers (calculating the overlap in the same mention type). Comparing this method with original overlap method, it efficiently reduce time used for comparing since we shrink the set needed to be compared by ignoring those tokens not have same mention type. Moreover, it dug deeper into the semantic meaning of the token rather than simply string matching, since only with the same mention type it is meaningful to compare these two mentions. Figure 3 is the evaluation part of scoring method using NameEntity Mention. We can see that this scoring method's precision is better than overlap scoring method (Figure 2) and this new scoring method's precision is the same with NGram method. However, the computation time of this new scoring method using NameEntity Mention is shorter than NGram scoring method. So until now I have implemented three scoring methods which means that the AnswerScore type has three different scores identified by different casProcessId.
There are also some trivial things we need to consider in realizing scoring using *NamedEntityMention*, such as I found that NameEntityMetion in Stanford NLP is extended from TOP rather than Annotation so when getting it from index we should not use the annotation index.

3.3 Deploy

In this part, I learned how to deploy my own UIMA-AS service and test that from the client. I think the difficulties in this part is the configuration things such as environment configuration and we also need to understand broker's job. I used it to deploy a service and it is necessary to keep it running when calling that service from client. Actually in this part I went rather smoothly.


Java - hw3-chenyinh-ziy-proto/src/main/resources/hw2-chenyinh-aae.xml - Eclipse
File Edit Navigate Search Project UIMA Run Window Help
Problems Javadoc Declaration Console Call Hierarchy History

```
New_configuration (2) [Java Application] /usr/lib/jvm/java-6-oracle/bin/java (Oct 7, 2013, 8:55:39 PM)
Oct 7, 2013 8:55:52 PM org.apache.uima.adapter.jms.client.BaseUIMAAsynchronousEngine_impl initializeProducer
INFO: Initializing JMS Message Producer. Broker: tcp://mu.lti.cs.cmu.edu:61616 Queue Name: ScnlpQueue
Oct 7, 2013 8:55:52 PM org.apache.uima.adapter.jms.client.BaseUIMAAsynchronousEngine_impl initializeConsumer
INFO: Initializing JMS Message Consumer. Broker: tcp://mu.lti.cs.cmu.edu:61616 Queue Name: ID:houcy-ThinkPad-T510-55393-1381193752127-0:0:1
Oct 7, 2013 8:55:53 PM org.apache.uima.adapter.jms.client.BaseUIMAAsynchronousEngine_impl initialize
INFO: Asynchronous Client Has Been Initialized. Serialization Strategy: [xmi] Ready To Process.
Booth shot Lincoln?
A 1 0.0 Booth shot Lincoln.
A 0 0.0 Lincoln shot Booth.
A 1 0.0 Lincoln was shot by Booth.
A 0 0.0 Booth was shot by Lincoln.
A 1 0.0 Booth assassinated Lincoln.
A 0 0.0 Lincoln assassinated Booth.
A 1 0.0 Lincoln was assassinated by Booth.
A 0 0.0 Booth was assassinated by Lincoln.
The precision is: 0.5
John loves Mary?
A 1 0.0 John loves Mary with all his heart.
A 1 0.0 Mary is dearly loved by John.
A 0 0.0 Mary doesn't love John.
A 0 0.0 John doesn't love Mary.
A 1 0.0 John loves Mary.
The precision is: 0.6666666666666666
Oct 7, 2013 8:55:54 PM org.apache.uima.adapter.jms.client.BaseUIMAAsynchronousEngineCommon_impl stop
INFO: Stopping Asynchronous Client.
Oct 7, 2013 8:55:54 PM org.apache.uima.adapter.jms.client.BaseUIMAAsynchronousEngineCommon_impl stop
INFO: Asynchronous Client Has Stopped.
Oct 7, 2013 8:55:54 PM org.apache.uima.adapter.jms.client.BaseUIMAAsynchronousEngineCommon_impl$SharedConnection destroy
INFO: UIMA AS Client Shared Connection Has Been Closed
Oct 7, 2013 8:55:54 PM org.apache.uima.adapter.jms.client.BaseUIMAAsynchronousEngine_impl stop
INFO: UIMA AS Client Undeployed All Containers
Average precision is: 0.5833333
```

Figure 3: Average Precision Score of NamedEntityMention Scoring Method

4 Conclusion and Discussion

From this homework, I first learned CPE with clear structure to split the whole processing part into different processes. Then through experiment I called and deployed the UIMA-AS service. Although I just used one client to call service but from building the client xml file I can see that AS support many different thread asynchronously using the AS mechanism, with a queue in front of each delegate, and which is really more flexible and more powerful compared with aae in hw2.

Comparing three different method to compute answer score(NGram and overlap scoring in hw2 and NameEntity Scoring in hw3), I found that in terms of the given input data sets, scoring method using NameEntity Scoring is best (considering both time execution and precision), Although the scoring method performance will depend on what kind of input document is.