10791 hw3 Report

Design and Engineering of Intelligent Information System

Jiacong He
Andrew ID: jiacongh

1.Introduction

This report mainly contains the following parts:
1. Create a CPE which is consiste of a Collection Reader, an Analysis Engine and a Cas Consumer
2. Create a UIMA AS client
3. Deploy my UIMA AS service
4. Run the StanfordCoreNLP annotator locally

2.Create and Running CPE(Task1)

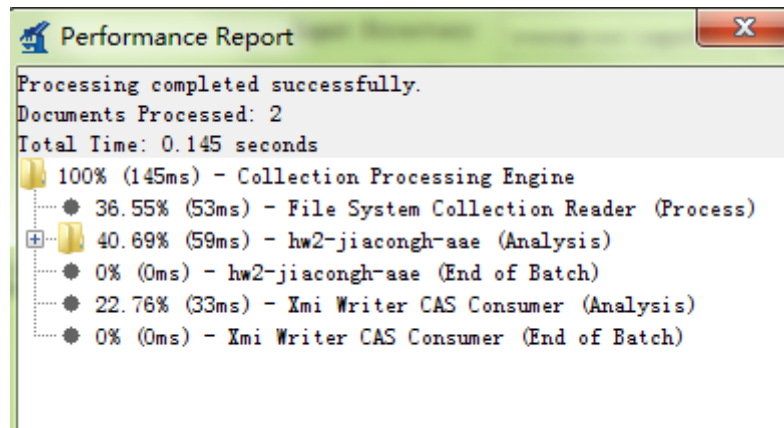In this task, I created a CPE and the construction is described as follow:

CPE :	hw3-jiacongh-aae-as-CPE.xml
Collection Reader:	FileSystemCollectionReader.xml
Analysis Engine:	hw2-jiacongh-aae.xml
CAS Consumer:	XmiWriterCasConsumer.xml

The FileSystemCollectionReader.xml file is designed to read the text file and convert it to a CAS object. Different from the Document Analyzer in hw2, the Collection Reader read the file for one time and will not read for several times.

The Analysis Engine is almost same with the one in hw2. But the evaluation part is removed.

The CAS Consumer is mainly the evaluation part in hw2. It help evaluation the scores of each sentence and print them on the console.
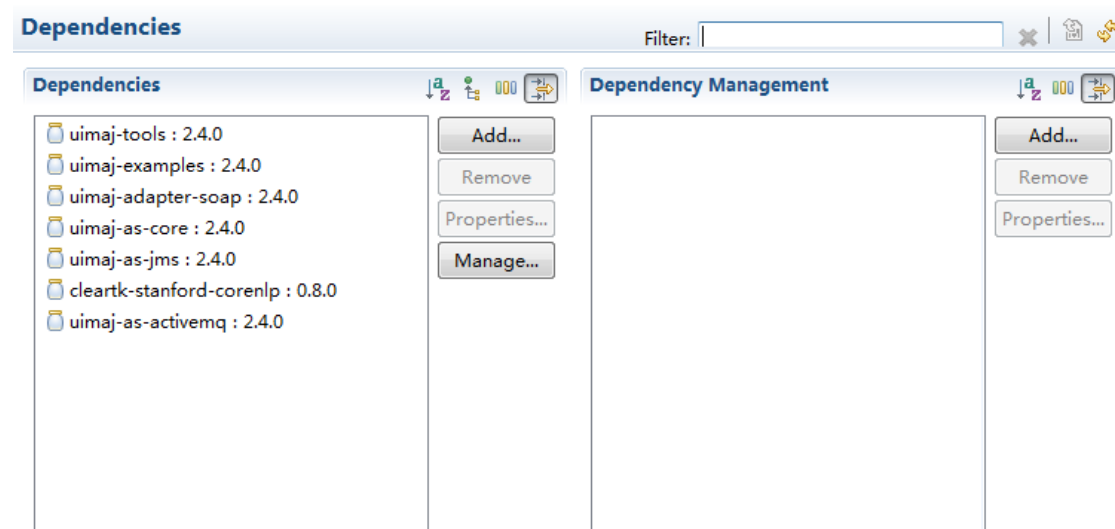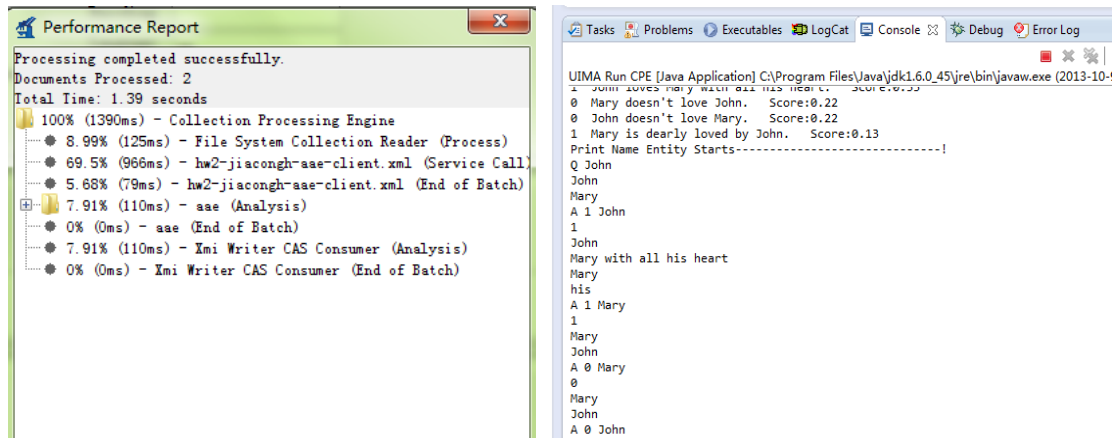
The result is shown as follow.

3.Deploy with UIMA AS(Task2)

In this task, I integrated a remote UIMA-AS service (Stanford CoreNLP) into my CPE pipeline, and deploy my aggregate analysis engine in homework 2 as an UIMA-AS service.
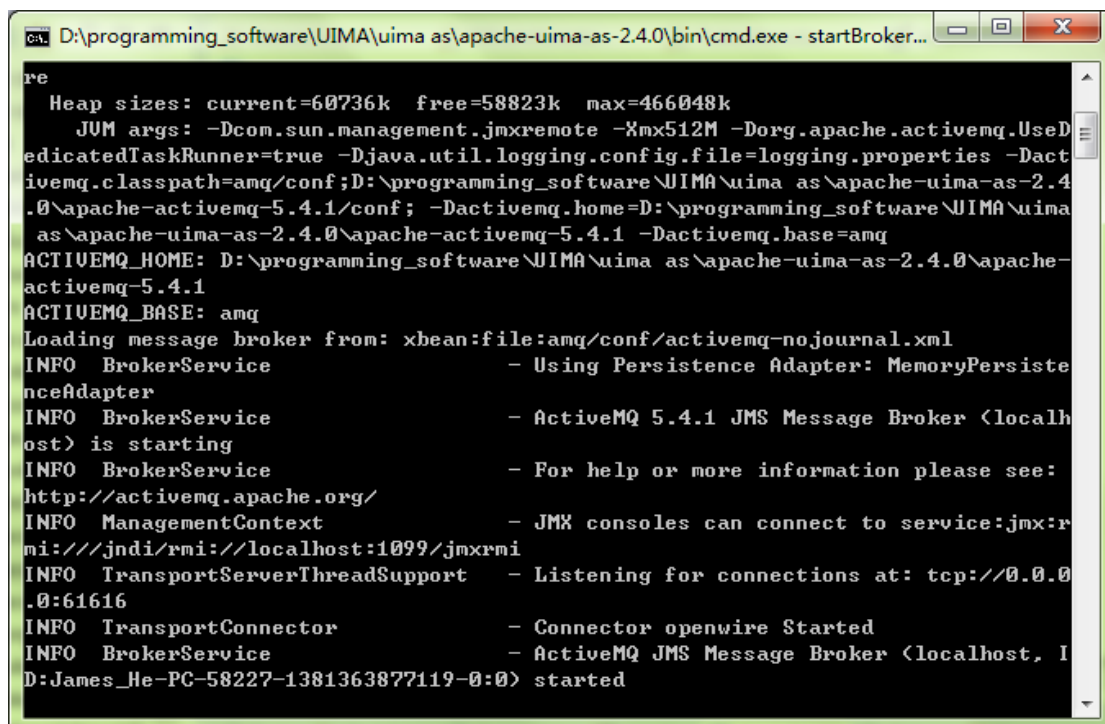
2.1 Creating an UIMA-AS client

In this task, I added the cleartk-stanford-corenlp and uima-as-activemq to the maven dependency. Then I add brokerURL and endpoint to the UIMA-AS service. Finally, I integrated the Name Entity annotations from the Stanford CoreNLP service into answer scoring component. I print all the name Entity out to console. The picture of this phase is as follow.

2.2 Deploying my UIMA-AS service

In this task, I deployed my aggregate analysis engine in Homework 2 on my machine, and called the service locally. I started the broker and use the connection tcp://0.0.0.0:61616 as the brokerURL in my aae-client

```
D:\programming_software\UIMA\uima as\apache-uima-as-2.4.0\bin\cmd.exe - deployAsy...

版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

D:\programming_software\UIMA\uima as\apache-uima-as-2.4.0\bin>deployAsyncService
.cmd C:\Users\James_He\git\hw3-jiacongh1\hw3-jiacongh\src\main\resources\descrip
tors\hw2-jiacongh-aae-deploy.xml
Service:SentenceAnnotator Initialized. Ready To Process Messages From Queue:inQ_
SentenceAnnotator_1.1
Service:QuestionAnnotator Initialized. Ready To Process Messages From Queue:inQ_
QuestionAnnotator_1.2
Service:AnswerAnnotator Initialized. Ready To Process Messages From Queue:inQ_An
swerAnnotator_1.3
Service:TokenAnnotaor Initialized. Ready To Process Messages From Queue:inQ_Toke
nAnnotaor_1.4
Service:NGramAnnotator Initialized. Ready To Process Messages From Queue:inQ_NGr
amAnnotator_1.5
Service:AnswerScoreAnnotator Initialized. Ready To Process Messages From Queue:i
nQ_AnswerScoreAnnotator_1.6
Service:EvaluationAnnotation Initialized. Ready To Process Messages From Queue:i
nQ_EvaluationAnnotation_1.7
Service:aae Initialized. Ready To Process Messages From Queue:ScnlpQueue
Press 'q'+'Enter' to quiesce and stop the service or 's'+'Enter' to stop it now.

Note: selected option is not echoed on the console.
```
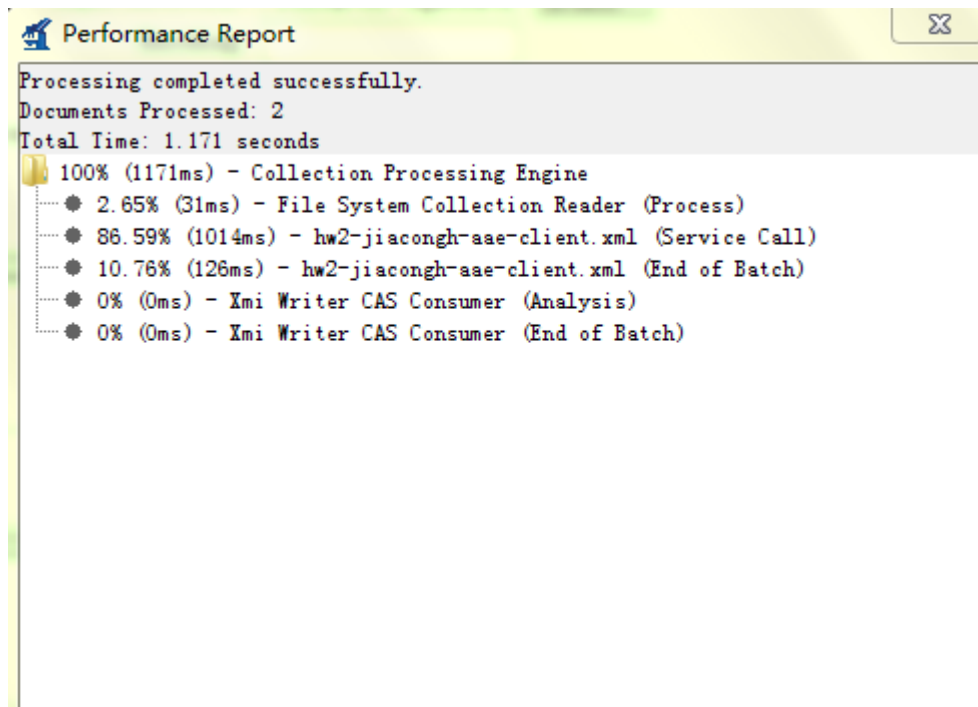
```
Tasks   Problems   Executables   LogCat   Console ⌧   Debug

UIMA Run CPE [Java Application] C:\Program Files\Java\jdk1.6.0_45\jre\bin\java

Question: [Booth shot Lincoln?]
Answers and Scores:
1  Booth shot Lincoln.   Score:1.00
0  Lincoln shot Booth.   Score:0.50
1  Booth assassinated Lincoln.   Score:0.33
0  Lincoln assassinated Booth.   Score:0.33
1  Lincoln was shot by Booth.   Score:0.25
0  Booth was shot by Lincoln.   Score:0.25
1  Lincoln was assassinated by Booth.   Score:0.17
0  Booth was assassinated by Lincoln.   Score:0.17

Question: [John loves Mary?]
Answers and Scores:
1  John loves Mary.   Score:1.00
1  John loves Mary with all his heart.   Score:0.33
0  Mary doesn't love John.   Score:0.22
0  John doesn't love Mary.   Score:0.22
1  Mary is dearly loved by John.   Score:0.13
```
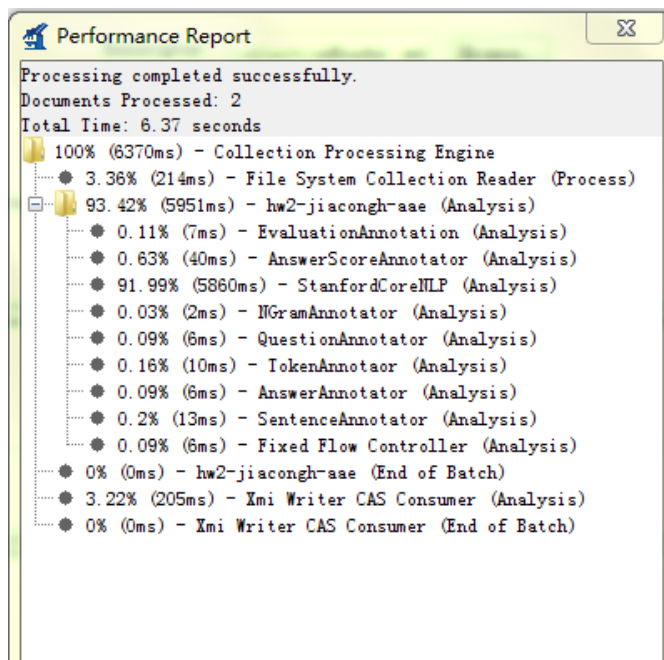
4. Run Stanford Annotator locally(Bonus)

In this part, I ran StanfordCoreNLP annotator locally. I added a StanfordCoreNLP Discriptor.xml which reflect to the StanfordCoreNLPAnnotator.java file.
The result is shown as follow:

```
Question: [John loves Mary?]
Answers and Scores:
1  John loves Mary.    Score:1.00
1  John loves Mary with all his heart.    Score:0.33
0  Mary doesn't love John.    Score:0.22
0  John doesn't love Mary.    Score:0.22
1  Mary is dearly loved by John.    Score:0.13
Adding annotator tokenize
Adding annotator ssplit
Adding annotator pos
Adding annotator lemma
Adding annotator ner
Adding annotator parse
Adding annotator dcoref
Print Name Entity Starts-----------------------------!
Booth
shot
Lincoln
```

From the picture we can find that the time of running locally is slower than calling remote service.