10791 hw2 report

Design and Engineering of Intelligent Information System

Andrew ID: jiacongh
Finished Date: 09/23/2013

1. Type System

1.1 Package Outline

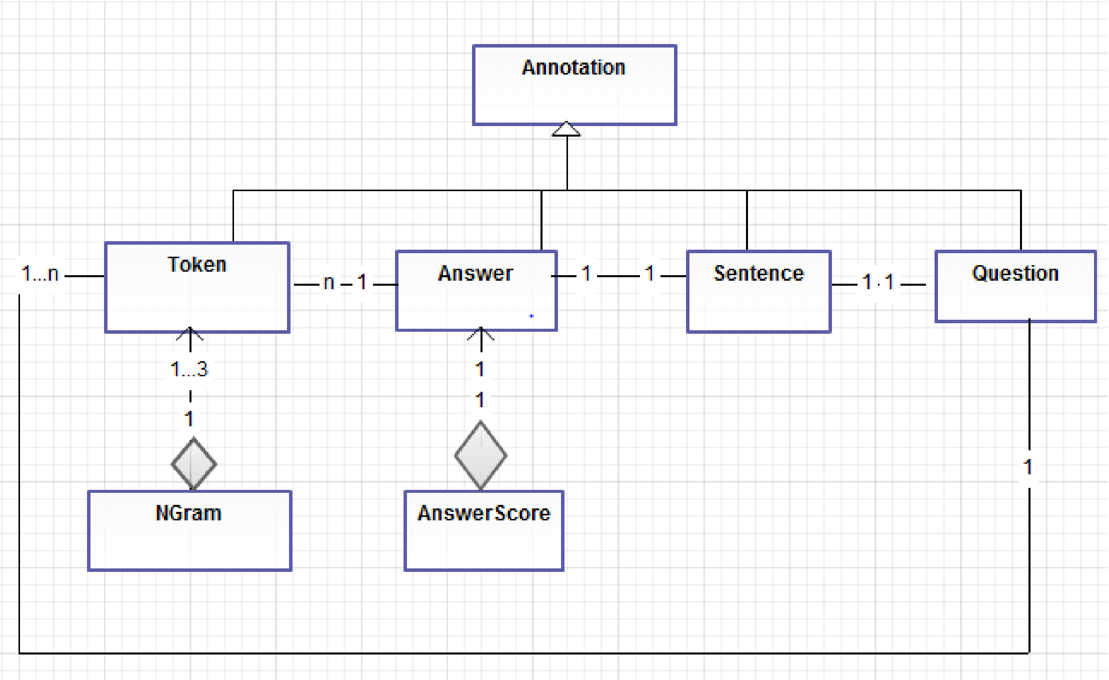
The overall type class is shown under in detail.

Annotation	Base class type of all annotations.
Annotation_Type	Types to assist save corresponding Annotation
Answer	Data structure to save all answers.
Answer_Type	Types to assist save corresponding Annotation
AnswerScore	Data structure to score for answers and make connection between answers and scores
AnswerScore_Type	Types to assist save corresponding Annotation
NGram	Data structure to save NGram words which is consisted of a list of tokens
NGram_Type	Types to assist save corresponding Annotation
Question	Data structure to save questions
Question_Type	Types to assist save corresponding Annotation
Sentence	Data structure to save sentence.
Sentence_Type	Types to assist save corresponding Annotation
Token	Data structure to save token.
Token_Type	Types to assist save corresponding Annotation

Each Annotation and its type can be seen as an integral part to save an annotated part from the document.

1.2 Hirarchical Structure

The hierarchical structure is presented in the picture under.



In this picture, I use one class to represent to java class – annotation and its type.
The Annotation class is the father class of all other classes. Among its subclasses, each Sentence may refer to on answer or one question. Each answer has 1…n Token and so is it with Question. Each NGram is composed from one, two or three tokens. And each AnswerScore is composed from one answer.

2. Annotator System

2.1 Package Outline

The overall type class is shown under in detail.

AnswerAnnotator	Annotator of Answer.
AnswerScoreAnnotator	Annotator of Scores of each answer.
EvaluationAnnotation	Last phase in the pipe line and sort each answer by their score and type them out
NGramAnnotator	Not used in this project
QuestionAnnotator	Annotator of Question.
SentenceAnnotator	Annotator of Sentence.

TokenAnnotator	Annotator of Token.

In this project, I wrote the NGram annotating action in the token directly to reduce the redundant token obtaining actions. Otherwise, it may need more time and actions to get tokens and their belonged questions and answers.

2.2 Function Description

The detail functions description is shown under representatively.

AnswerAnnotatior:

void	**initialize**(org.apache.uima.UimaContext aContext) Initialize parameters and assign pattern regex its corresponding value.
void	**process**(org.apache.uima.jcas.JCas aJCas) Recogonize the answer if a sentence start with word "A".

AnswerScoreAnnotator

java.util.ArrayList[]	**getAswNGramList**(org.apache.uima.jcas.JCas aJCas) Find NGrams that belongs to the answer and returns a list of NGram objects.
java.util.ArrayList <NGram>	**getQstNGramList**(org.apache.uima.jcas.JCas aJCas) Find NGrams that belongs to the question and returns a list of NGram objects.
void	**process**(org.apache.uima.jcas.JCas aJCas) Get all answers and their corresponding NGrams.

EvaluationAnnotation

void	**process**(org.apache.uima.jcas.JCas aJCas) Get all AnswerScore objects, sort them by score and print the sorted documents to the console.

NGramAnnotator

void	**process**(org.apache.uima.jcas.JCas aJCas)

	Not used in this project

QuestionAnnotator

void	**process**(org.apache.uima.jcas.JCas aJCas) Get each Sentence object and analyze it If contain "Q" then put the text into Question object

SentenceAnnotator

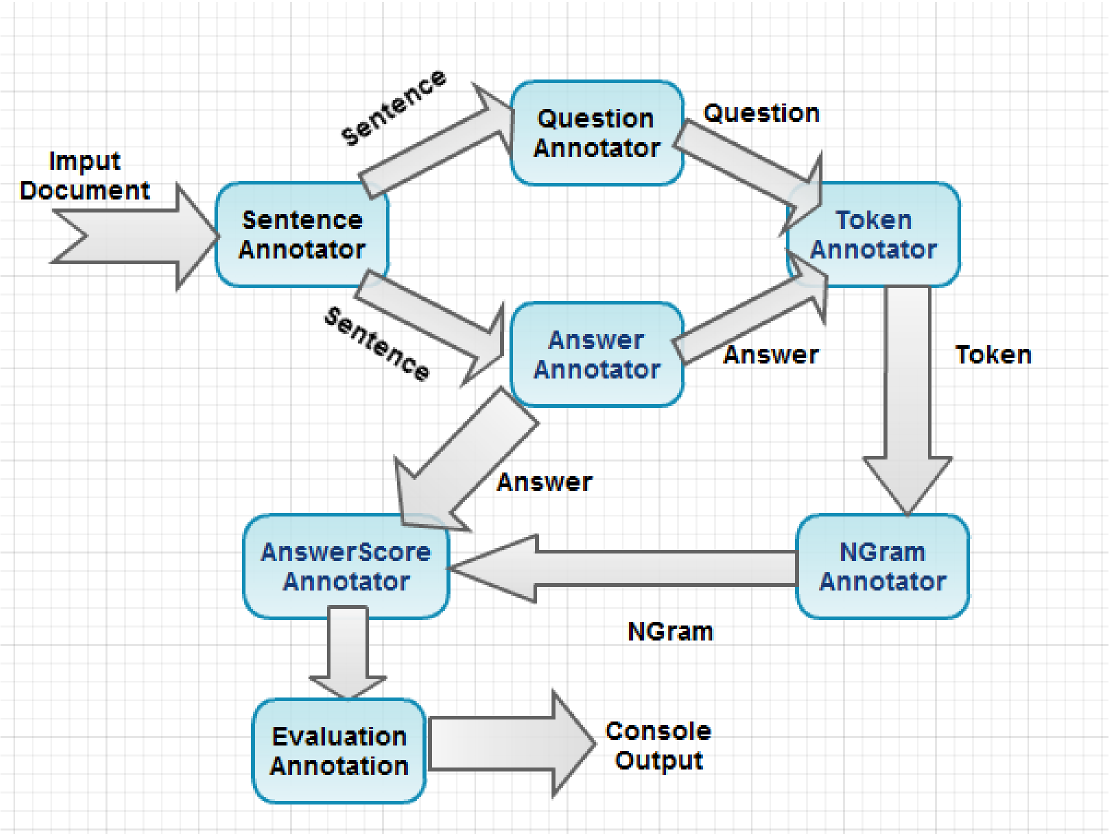
void	**process**(org.apache.uima.jcas.JCas aJCas) Analyze the document by each line and cut out its line break

TokenAnnotator

void	**initialize**(org.apache.uima.UimaContext aContext) initialize some parameters and assign some pattern regix to the puncPattern regix.
void	**process**(org.apache.uima.jcas.JCas aJCas) Cut out punctuation in sentence, split sentence with blanks and use Token object to save each token.

3.Design Pattern

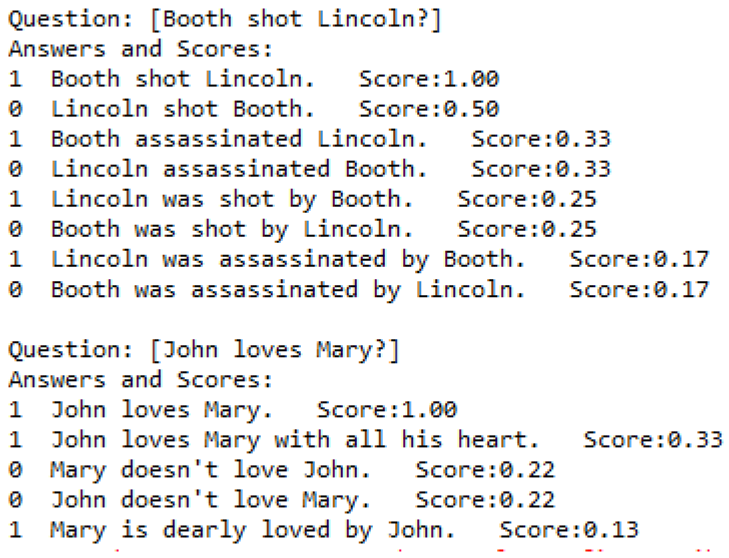
In this project, I use a hierarchical method to construct analysis engine and combine them into a pipeline. In the pipeline, the first AE is the SentenceAnnotator. It create one Sentence object each time recognizing a line break. Then the QuestionAnnotator and AnswerAnnotator both process with the result from SentenceAnnotator. Then the TokenAnnotator is used to recognize each token and I also put the NGram recognition methed here to recognize NGram directly from the token result. After that, the AnswerScore is used to confirm each answer’s score. Finally, I used the EvaluationAnnotation to rank the answers by their score and output them. The figure under can better describe this process.



4.Result

4.1 Console Output

Console output contains the question, the ranking of answers, answers’ score and the standard result of each answer sentence. The result picture is shown as follow.

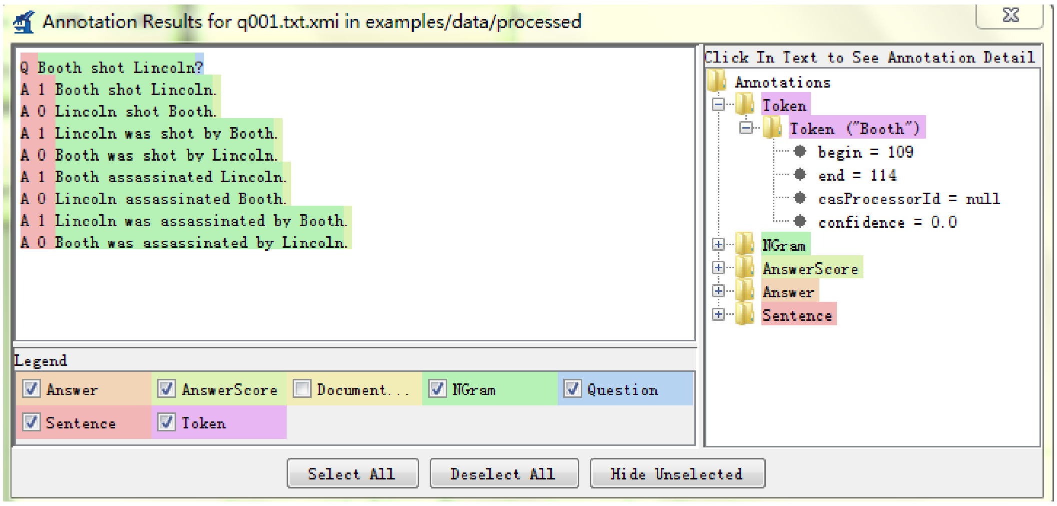


```
Question: [Booth shot Lincoln?]
Answers and Scores:
1  Booth shot Lincoln.   Score:1.00
0  Lincoln shot Booth.   Score:0.50
1  Booth assassinated Lincoln.   Score:0.33
0  Lincoln assassinated Booth.   Score:0.33
1  Lincoln was shot by Booth.   Score:0.25
0  Booth was shot by Lincoln.   Score:0.25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

4.2 GUI Result

In the GUI result, I label the Answer, AnswerScore, DocumentAnnotation, NGram,Question, Sentence and the Tokens. The result is shown in the under picture.



Annotation Results for q001.txt.xmi in examples/data/processed

```
Q Booth shot Lincoln?
A 1 Booth shot Lincoln.
A 0 Lincoln shot Booth.
A 1 Lincoln was shot by Booth.
A 0 Booth was shot by Lincoln.
A 1 Booth assassinated Lincoln.
A 0 Lincoln assassinated Booth.
A 1 Lincoln was assassinated by Booth.
A 0 Booth was assassinated by Lincoln.
```

Click In Text to See Annotation Detail

```
Annotations
  Token
    Token ("Booth")
      begin = 109
      end = 114
      casProcessorId = null
      confidence = 0.0
  NGram
  AnswerScore
  Answer
  Sentence
```

Legend

☑ Answer ☑ AnswerScore □ Document... ☑ NGram ☑ Question

☑ Sentence ☑ Token