11-791 Design and Engineering of Intelligence Information System
Homework 1

Logical Data Model and UIMA Type System Design & Implementation

Lab Report

Andrew id: xchu

1 General Description

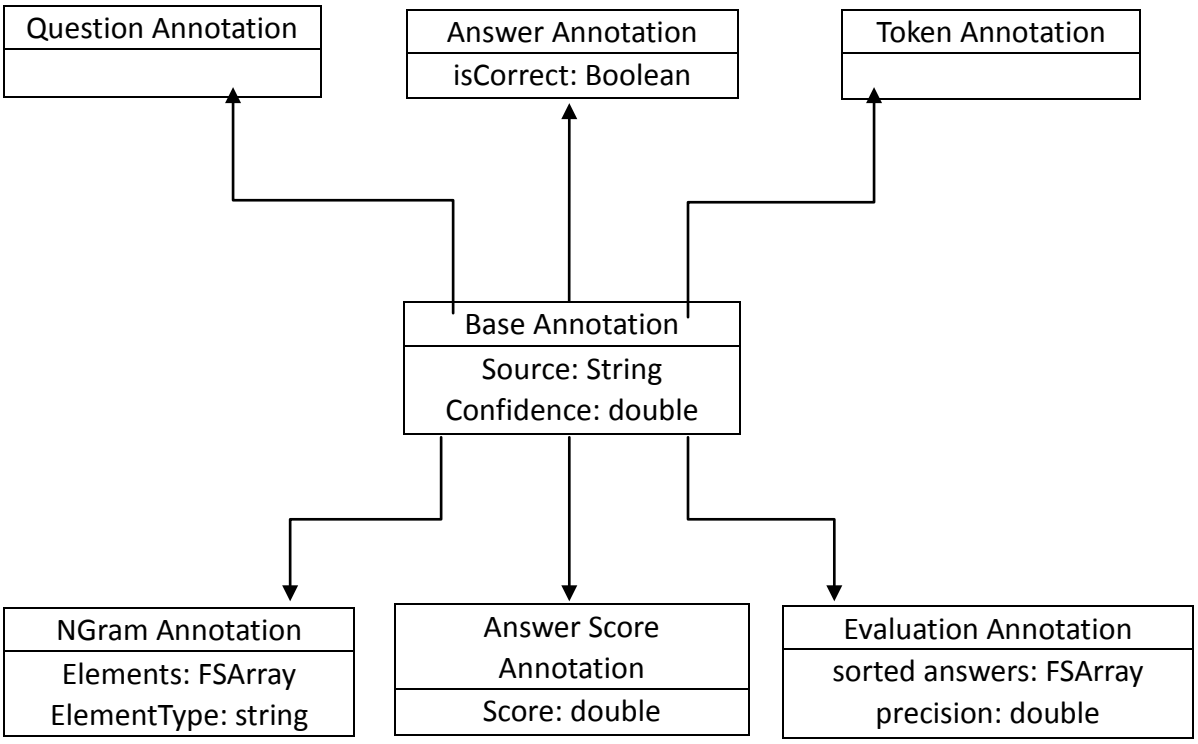
In this lab, I have created a logical data model for a sample information processing system. Given a question and a set of questions, we first read the input file and annotate the questions and answers. Then we assign a score to each sentence, rank the sentences according to scores, select the top N sentences where N is the number of correct answers and finally measure the precision score by N. The whole project was implemented on the base of UIMA system. I first analyzed the system requirements and then design an appropriate UIMA type system to model the required information types. By using UIMA tooling in Eclipse, I created the type system.xml file and use the JCasGen plugin to compile the type system into Java type classes.

2 Design Pattern

For the whole type system, I first create a base annotation type including two features: a string feature source and a numeric feature confidence. All other types inherit from this base annotation type.
From the pipeline requirement analysis, I know at the beginning of information system, we need to design two annotations: question and answer annotation. The system will read in the input file as a UIMA CAS and annotate the question and answer spans. Meanwhile, we also need a Token annotation. This will annotate each token span in each question and answer. In this system, I also designed a NGram Annotation, this will annotate 1,2,3-grams of consecutive tokens.
For the final part of this system, as it will incorporate a component that will assign an answer score to each answer, we also need an answer score annotation. The last annotation we need to design is evaluation annotation which will sort the answers according to their scores and calculate precision at N.

3 Type System

For the type system, their inherit relationship can be described as below:



The function of every annotation is:
Base Annotation:
It is the base annotation that all other annotation inherits. It has two features, the first is source to indicate where this annotation comes from. The other feature is confidence to keep track of how confidence the annotation is

Question Annotation:
The system will read input file and annotate the question spans. So in this annotation we can get the results for the question.

Answer Annotation:
It is similar to question annotation, except that it can annotate whether the answer is correct or not.

Token Annotation:
Token annotation inherits from base annotation. The system will annotate each token in each question and answer.

NGram Annotation:
Our system will annotate 1-, 2- and 3-grams of consecutive tokens. This annotation have two more features. First is elements which will annotate the result of any given N-grams. The other is element type.

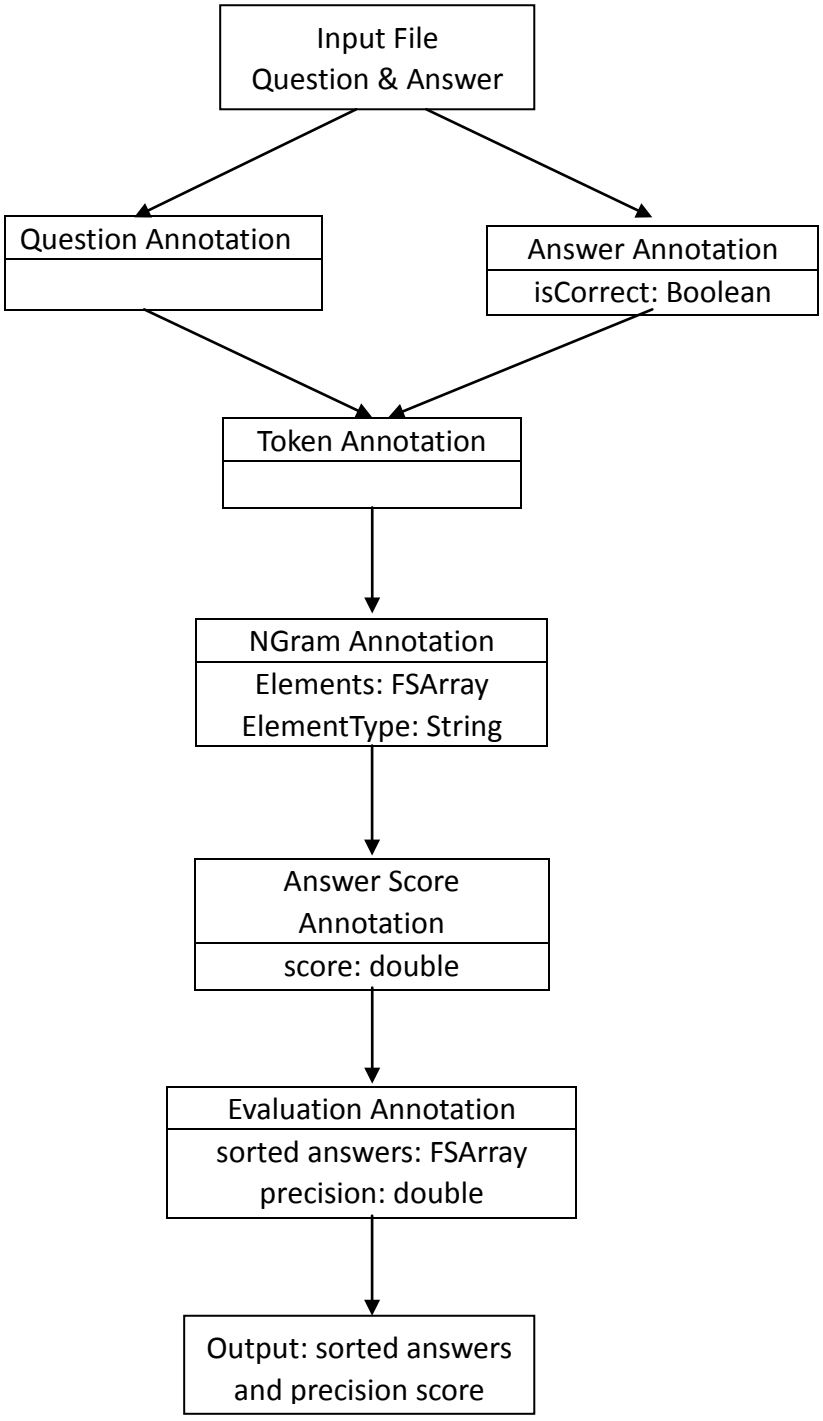
Answer Score Annotation:

The system will assign an answer score annotation to each answer. This annotation will record the score assigned to the answer.

Evaluation Annotation:
The system will sort the answers according to their scores and calculate precision at N. So we need this annotation to record the answers precision score.

4 UML Design

For the whole information processing system, the UML can be described as below:



5 Engineering Issues:

(1) Annotator

In this project, I also designed two Annotators. First is RegexAnnotator, the second is NGram Annotator. Based on these steps, we can set up an Analysis Engine Descriptor. In the parameter settings, we can set the value of each parameter and then get their value while executing the programs. Note I have encountered a problem on casting when I first did this task, it indicates that I cannot pass the value to the string variables in my code. Finally, the real reason is that in the parameter settings, I choose the multi-value, so the program will pass the String[] which cannot cast to string. After I reset them to Single-value, the problem was solved.

(2) Java comments

We can generate the specific Java comments by JCasGen in the type system.xml. However, this may not enough. According to the oracle java comments, we can also add some other information in the Java classes after JCasGen. In this project, I also included some other information like “@see” ,”@version”, @”param” in the generated methods.