11791

Design and Engineering of Intelligent Information Systems

PI2: UIMA Type System

Yiu-Chang Lin
yiuchanl@cs.cmu.edu

1. System Design Analysis

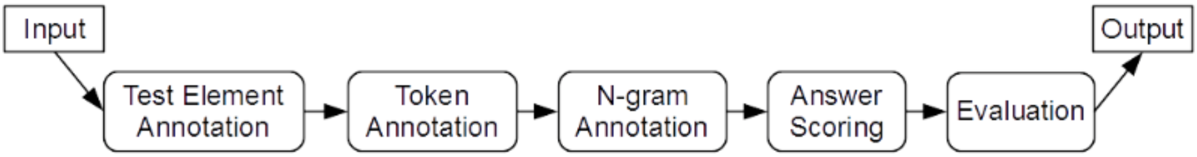


Fig. 1

The high level architecture of the system is shown in Figure 1. The raw input text consists of one question and five corresponding answers with label 0 or 1. The text is first annotated by a token annotator and then by a N-gram annotator. Afterwards, answers are scored by N-gram matching and the result is evaluated by the metric precision @ N. Therefore, it is straightforward to come up with the following types in mind to design the whole system:

-Token
-N-Gram
-Question
-Answer
-Evaluator

2. UIMA Type System

Figure 2 and Figure 3 show the UML class diagram of my UIMA type system. In the following subsections, each type will be described in detail.

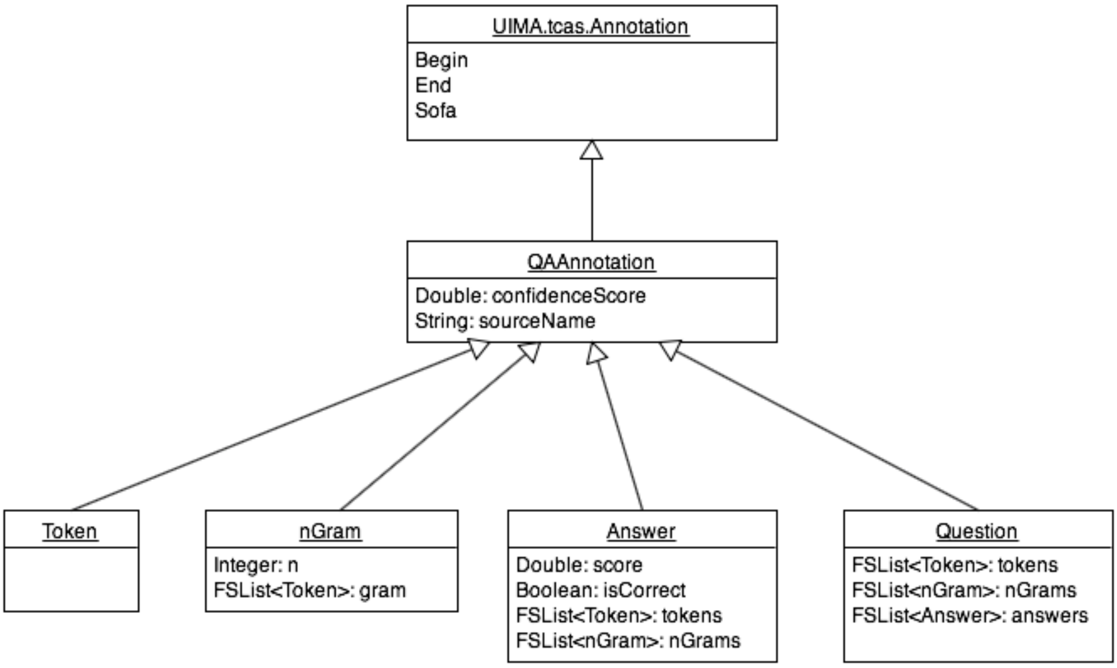
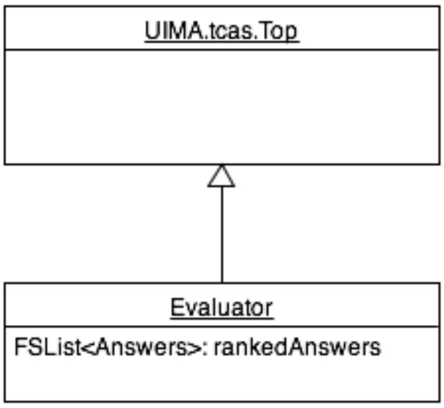


Fig. 2



Fug. 3

2.1 *QAAnnotation*

QAAnnotation, inherited from *UIMA.tcas.Annotation*, is the super class for all annotation classes. It requires two fields, *confidencScore* and *sourceName*. The former is how confidence it is annotated and the later is name of the source that generates the annotation.

2.2 Token

Type *Token* only needs begin and end features so it has no additional fields.

2.3 nGram

Instead of creating three different types for uni-gram, bi-gram and tri-gram, respectively, I decide to use a more flexible way to implement n-gram. Field *n* indicates the degree of n-gram and field *gram* stores the list of tokens corresponding to n-gram.

2.4 Answer

The *score* feature stores the n-gram score of this answer. The *isCorrect* feature is true if the label of this answer is correct and false if incorrect. Field *tokens* stores the tokenization result of the answer in a list of *Token* and *nGrams* store the n-gram result in a list of *nGram.*

2.5 Question

The difference between *Question* and *Answer* is the former does not have *score* and *isCorrect* features that are specific to *Answer.* In contrast, it has a list of *Answer,* which stores the corresponding answer to this question.

2.6 Evaluator

Evaluator is inherited from *UIMA.tcas.TOP* since it is not an annotation. The functionality of *Evaluator* is to calculate the precision @ N score for the system using its feature, *rankedAnswers*, which is a list of its answers ranked by score.

3. Summary

In this project, we first started from analyzing the system architecture, reading UIMA tutorial and at last designing our own type system. It is a good practice to go through this whole process by implementing a Logical Data Model for a sample information processing task. The UIMA type system is only the first step and I really look forward to the following projects to make this pipeline work on real input data.