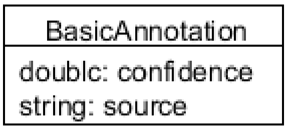
11-693 Software Methods for Biotechnology Homework1 Report

Shen Wu

1. System design:

1.1. Basic annotation type:

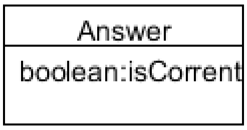
Followed the instruction, I designed my own basic annotation type inherited from annotation type. I added two features: confidence and source. All other types created later will be inherited from the basic annotation type so that they all have the feature of confidence and source.

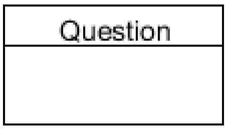


1.2. Element type:

Two types are designed to annotate the whole document: Question and Answer.

Answer type has the feature : isCorrent, which is decided by the gold answer provided by the input doc.



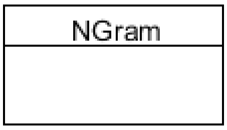


1.3. Token annotation:

Token type was created to annotate tokens.

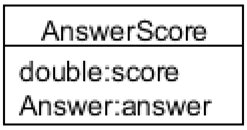
1.4. N-Gram annotation:

UniGram, BiGram and TriGram types are designed, and also a type named NGram is created to record all these NGram tokens. Feature named elements which is FSArray of Token type is defined to record elementary tokens of each NGram token.



1.5. AnswerScore

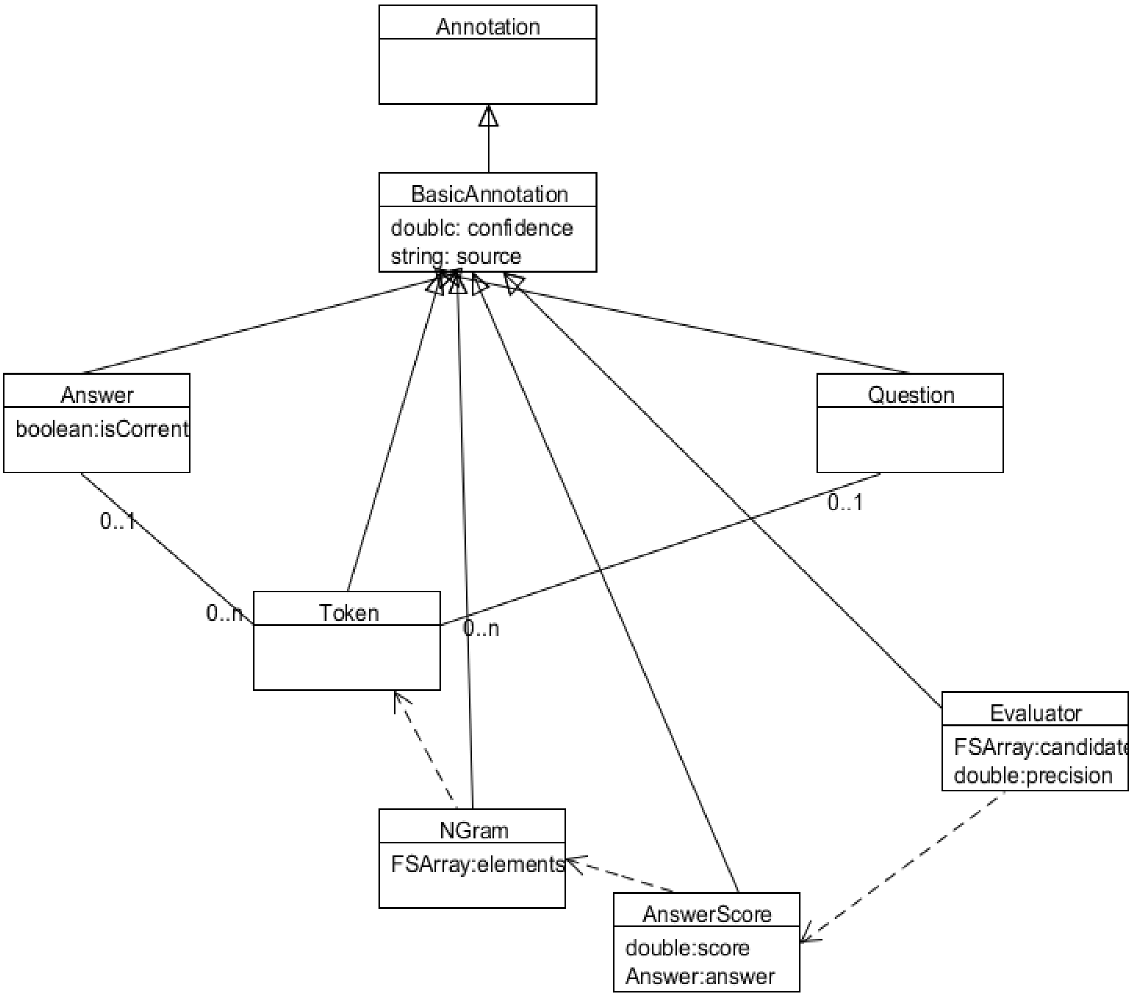
an AnswerScore type is used to record (annotate) the score of each answer sentence.



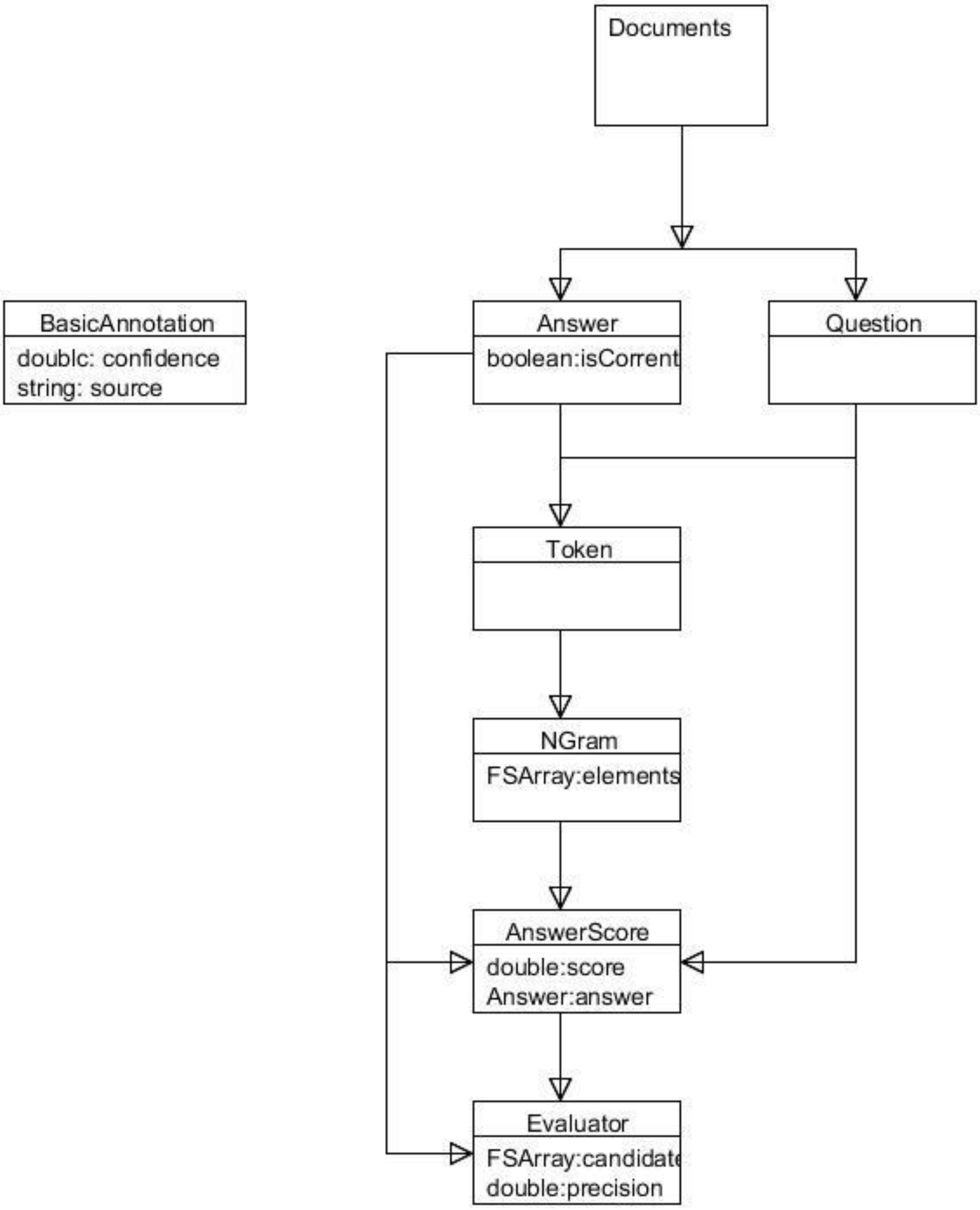
1.6. Evaluator

Evaluator is the type which will record sorted answers according to their scores, and calculated precision at N (where N is the total number of correct answers).

2. UML description of class diagram:



3. UML description of flow-process diagram:



4. Annotators

Following annotators are designed to annotate documents

4.1. ElementAnnotator

Input:

Output: Answer, Question

4.2. TokenAnnotator

Input: Answer, Question
Output: Token

4.3. UniGramAnnotator
Input:Token
Output:NGram

4.4. BiGramAnnotator
Input:Token
Output:NGram

4.5. TriGramAnnotator
Input:Token
Output:NGram

4.6. AnswerScoreAnnotator
Input:Answer, NGram
Output:AnswScore

4.7. EvaluatorAnnotator
Input:Answer, AnswerScore
Output:Evaluator

5. comparison of other methods

Since in this homework, the annotators are not required to implement, the comparison will be focus on the different design.

An alternative design is not using types such as Token and NGram, extract tokens and calculate score at the same time. The system will be more concise this way. But in the long run, this kind of design is not easy to be improved or extended.