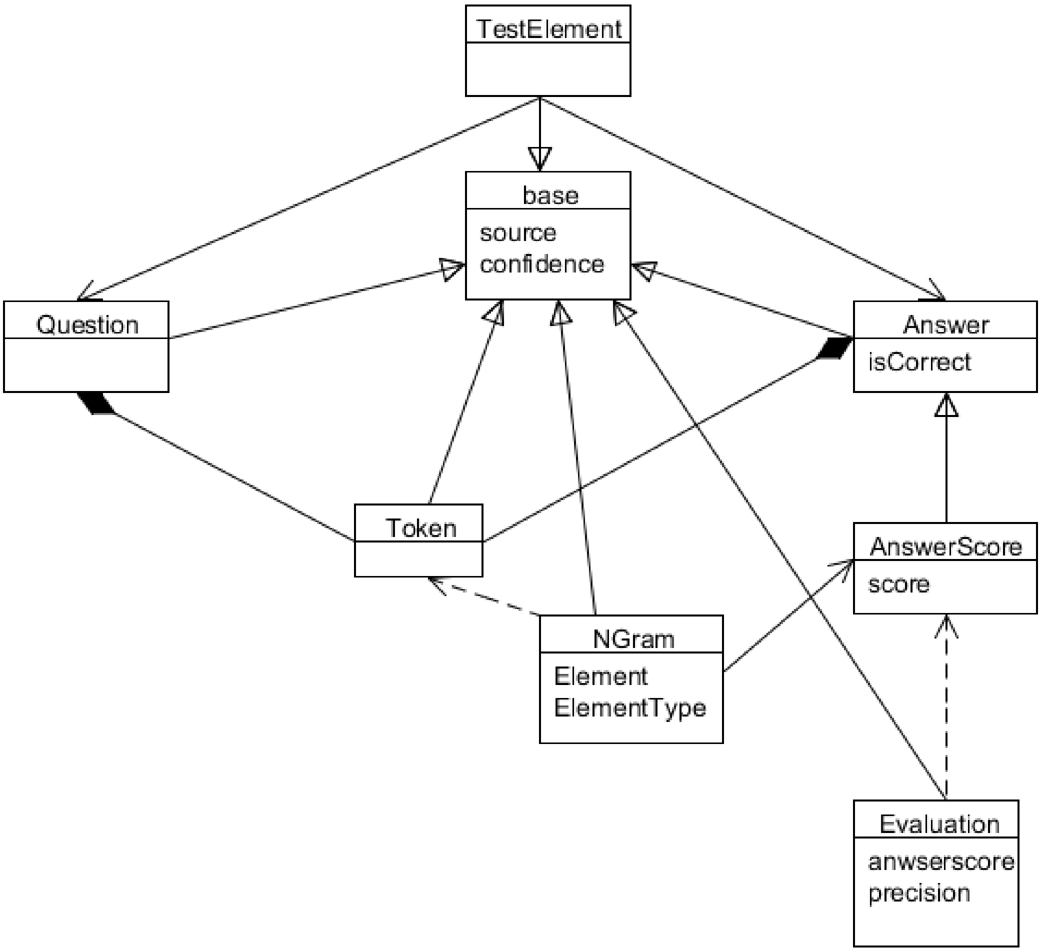
Report

Qiwen Zhu

Then, the first diagram shows the typeSystem I designed.
All the type are automaticly given the features *'begin'* and *'end'*, so I do not show them in the picture.
The type ***'base'*** includes two features: '*source'* (string) and '*confidence'*(double). They "help to keep track of where an annotation was originally made by, and how confidence the annotation was All other types then inherit from this base annotation type."(cited from the hw1 instructor).
The type 'NGram' has two particular features:'Element' and 'ElementType'. Feature 'Element' is used to store and deal with token. I also plan to implement 1-、2-、3-Gram within 'NGram'.



The second diagram is the flow chart of the program. It describes the processing pipeline cited from the hw1 instructor:

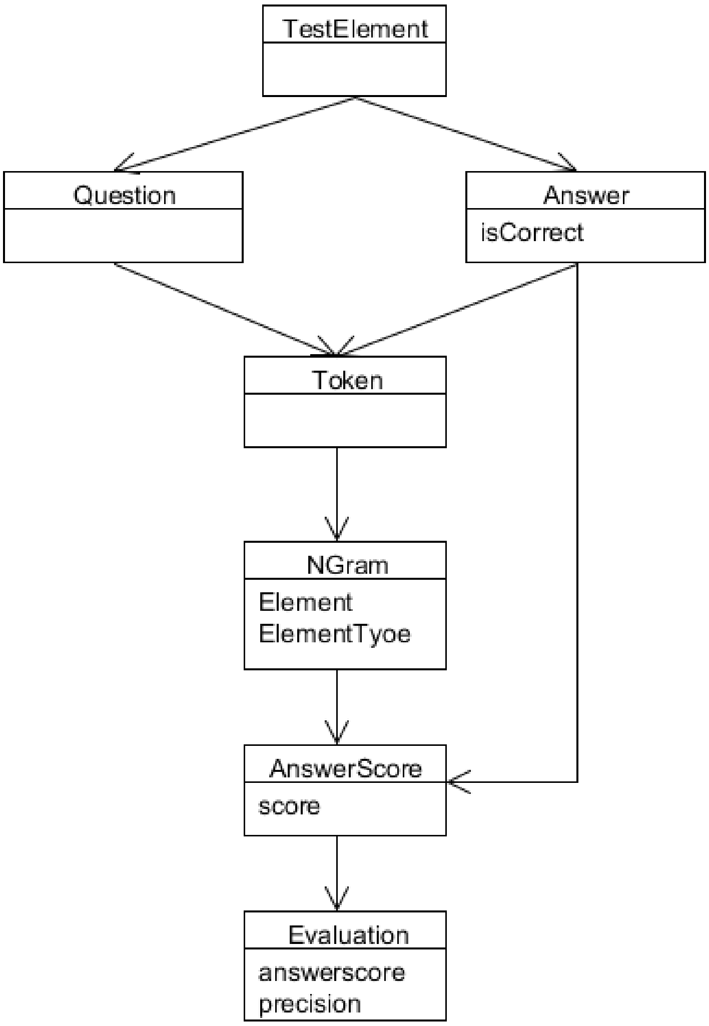
"1. Test Element Annotation: The system will read in the input file as a UIMA CAS and annotate the question and answer spans. Each answer annotation will also record whether or not the answer is correct.

2. Token Annotation: The system will annotate each token span in each question and answer (break on whitespace and punctuation).

3. NGram Annotation: The system will annotate 1-, 2- and 3-grams of consecutive tokens.

4. Answer Scoring: The system will incorporate a component that will assign an answer score annotation to each answer. The answer score annotation will record the score assigned to the answer.

5. Evaluation: The system will sort the answers according to their scores, and calculate precision at N (where N is the total number of correct answers)."


TestElement
Question
Answer
isCorrect
Token
NGram
Element
ElementTyoe
AnswerScore
score
Evaluation
answerscore
precision