Homework 1 Report

Ran Zhao Sep 10, 2013

Part I. Pipeline Overview

In this pipeline, there are seven phases based on the required steps.

Steps:

1. Test Element Annotation (Question and Answer Annotation)
2. Token Annotation
3. NGram Annotation
4. Answer Scoring
5. Evaluation

Pipeline:

FileCollectionReader → Tokenizer → Segmentator → Question&Answer Parser →

Answer Score Generator → Evaluation → ResultConsumer

(1) FileCollectionReader: It is a file reader that read all the files as input under a directory.

(2) Tokenizer: It will tokenize the input file and output the token annotation. There are several NLP tools available such as OpenNLP.

(3) Segmentator: It will segment the file into NGram annotations.

(4) Question&Answer Parser: It will parse the text into question and answers. The whole fields of the question annotation should be filled. However, the fields of the answer annotation will be filled except score. The fields of the Unit annotation will be filled except answerList and precision.

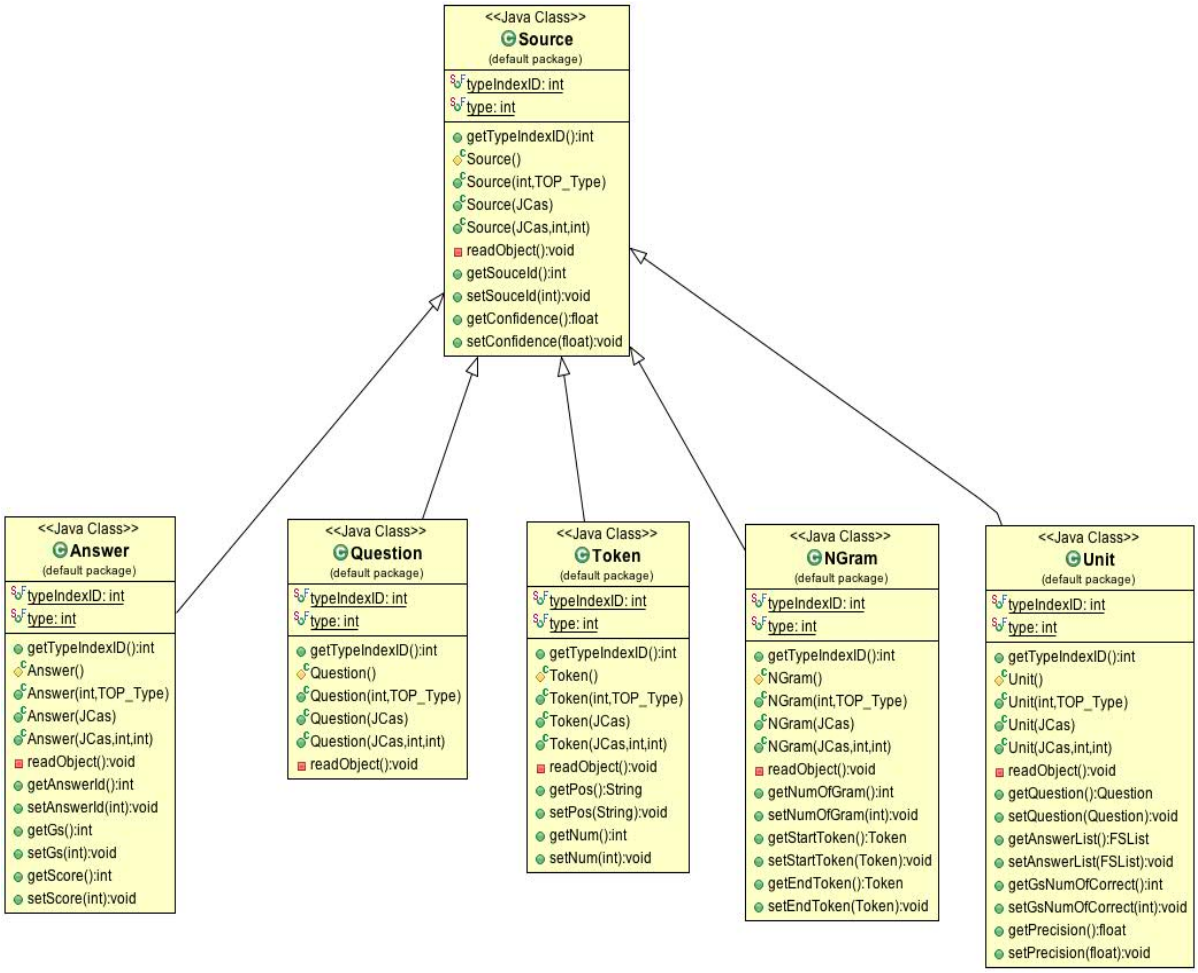
(5) Answer Score Generator: It will assign an answer score to each answer. The score field of the question annotation should be filled.

(6) Evaluation: It will sort the answer according to their scores and calculate precision at N. The answerList and precision fields of Unit annotation should be filled.

(7) ResultConsumer: It will output the annotations into files.

Part II. Type System Design & Implementation

When I design the type system, firstly, I should meet the annotation requirement. Thus, I design my type system in an inheritance way. Of course, the base type is uima.tcas.annotation. Then I create a base annotation type "source" to inherit from uima.tcas.annotation type. All the other types a all inherited from source type. In order to clarify the hierarchical relations between types, I draw an UML diagram.



(1) *Type:* Source

---It is a base type of the system and inherited from uima.tcas. annotation.

Feature:

- sourceId: It indicates the id of the resource. Each question and answers pair has a unique id.
- confidence: It indicates the confidence of the annotation.

(2) *Type:* Answer

---It is answer annotation which is inherited from source type.

Feature:

- answerId: It indicates the # of the answer to the question.
- gs: It indicates the golden standard records of the answer.
- Score: It is a score assigned by the answer score generator.

(3) *Type:* Question

---It is question annotation which is inherited from source type.

(4) *Type:* Token

---It is a token annotation which is inherited from source type.

Feature:

- pos: It indicates the part of speech of the token which may be used later.
- num: It indicates the # of the token in a unit (a question and its answers = a unit)

(5) *Type:* NGram

---It is NGram annotation which is inherited from source type.

Feature:

- numOfGram: It indicates the # of the grams.
- startToken: It indicates the start token of a segment.
- endToken: It indicates the end token of a segment.

(6) *Type:* Unit

---A question and its answers = a Unit.

Feature:

- question: It indicates the question in this unit.
- answerList: It is a sorted order answer list based on the score.
- gsNumOfCorrect: It is the total number of correct answers in golden standard records.
- precision: It is the precision of the unit.

Part III. Example

(1) Phase1: FileCollectionReader:
Input: q001.txt
Output:
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.

(2) Phase2: Tokenizer
Input: Text
Output:
Ex. Lincoln,was,shot,by,Booth,…

(3)Segmentator:
Input: Text and tokens
Output:
Ex. N=1 Lincoln<numOfGram=1><startToken=Lincoln><EndToken=Lincoln>,…
N=2 Lincoln was<numOfGram=2><startToken=Lincoln><EndToken=was>,…
N=3 Lincoln was shot<numOfGram=3><startToken=Lincoln><EndToken=shot>,…

(4)Question&Answers Parser:
Input: Text
Output:
Question Annotation: Booth shot Lincoln?
Answer Annotation: Booth shot Lincoln <gs=1> <score=null>, …
Unit Annotation: <question= Booth shot Lincoln >
<answerList=null><NumOfCorrect=4><precision=null>

(5)Answer Score Generator:
Input: Text and Answer Annotation
Output:
Answer Annotation: Booth shot Lincoln<gs=1><score=1>,…

(6)Evalutaion:
Unit Annotation: <question= Booth shot Lincoln > <answerList= Booth shot Lincoln, Lincoln was shot by Booth,… ><NumOfCorrect=4><precision=0.75>

Note: As source is the base type of the system, I did not write the sourceId and confidence to each type. However, each of them should have these two features. In each phase, I only show the updated annotation rather than the whole result in Cas.