11-791 HW1: Logical Data Model and UIMA Type System Design & Implementation

Xiaohua Yan

Language Technologies Institute

The Processing Pipeline

This document describes the design and implementation details of the UIMA type system for a sample information processing system, whose pipeline consists of the following 5 steps:

1. **Test Element Annotation**: The system reads 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 annotates each token span in each question and answer (break on whitespace and punctuation).
3. **N-gram Annotation**: The system will annotate 1-, 2- and 3-grams of consecutive tokens.
4. **Answer Scoring**: The system incorporates 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 sorts the answers according to their scores, and calculate precision at N (where N is the total number of correct answers).

Type System Implementation

This sections reports the implementation details of the type system according to the requirement of the information processing system.

Base Types

Since the processing pipeline asks for various kinds of annotation for the same *subject of analysis*, it would useful to keep track of where an annotation or evaluation result originally came from, and how confidence the (annotation) result is. Therefore I define two *base types* as illustrated in the following table.

Type Name	Super Type	Description
BaseAnnot	Annotation	Base type for other annotation types
BaseTOP	TOP	Base type for other TOP types

Both of the base types have two features named $srcComponentId$, which records the source of the annotation or other results, and $confidence$, which measures how confidently the result is produced.

Test Element Annotation

The test element annotator of the system should annotate the question and all the corresponding answers in the input file. In my type system, two annotation types named *Question* and *Answer* are created for each question and each answer respectively, which both inherit from the *BaseAnnot* type (same for all other annotation types). And for the *Answer* annotation type in particular, the boolean feature *isCorrect* is created to record whether the answer is correct or not.

Token Annotation

Another annotation type named $Token$ is created for each token in the document. Notice that though I did not add extra features by now for the $Token$ type, it is possible that other features of tokens such as POS tags can be useful for more complicated tasks.

N-gram Annotation

The annotation type named $NGram$ is defined to annotate n-grams of consecutive tokens. Note that I did not define features to record the order of n-grams but instead define the type of content of n-grams to be $FSArray$ whose elements are $Token$s such that the annotator is not limited to 1-, 2- and 3-grams as suggested in the handout.

Answer Scoring

The *AnswerScore* annotation type is defined to annotate the score of each answer given by the corresponding scoring component. A feature named *answer* is also declared to keep track of the corresponding string of answer.

Evaluation

For the evaluation part of the system, the type *EvaluationResult* is created, which has 3 features named *precision*, *sortedAnswers* and *numberOfCorrectAnswers* respectively that serve as the output of the system. To be specific, *precision* denotes the precision at *numberOfCorrectAnswers* which denotes the total number of correct answers and *sortedAnswers* is created as an FSArray, the type of whose elements is *Answer*.

Other Types

Besides all the types described above, the annotation type *Document* is defined to annotate the input or output document, with *docName* as its feature that records the name of the document file.

Type Definition in Details

Finally, the following table illustrates the definition of all the types I defined as well as their super types.

Type Name	Super Type	Description	Features	Range of Features
BaseAnnot	Annotation	Base type for other annotation types	srcComponentId confidence	String Double
BaseTOP	TOP	Base type for other TOP types	srcComponentId confidence	String Double
Document	BaseAnnot	Annotation type for each document	docName	String
Question	BaseAnnot	Annotation type for each question		
Answer	BaseAnnot	Annotation type for each answer	isCorrect	Boolean
AnswerScore	BaseAnnot	Annotation type for answer scores	answer	Answer
Token	BaseAnnot	Annotation type for each type		
NGram	BaseAnnot	Annotation type for each n-gram	elements	FSArray of Token
EvaluationResult	BaseTOP	Type for evaluation results	precision sortedAnswers numberOfCorrectAnswers	Double FSArray of Answer Integer