11791 Design and Engineering of Intelligent Information Systems

Fall 2013 Assignment 2

Kuo Liu
kuol@andrew.cmu.edu

September 10, 2013

Logical Architecture and UIMA AAE Design

Figure 1 illustrates the overall data flow that occurs among the different components which make up the Aggregated Analysis Engine.

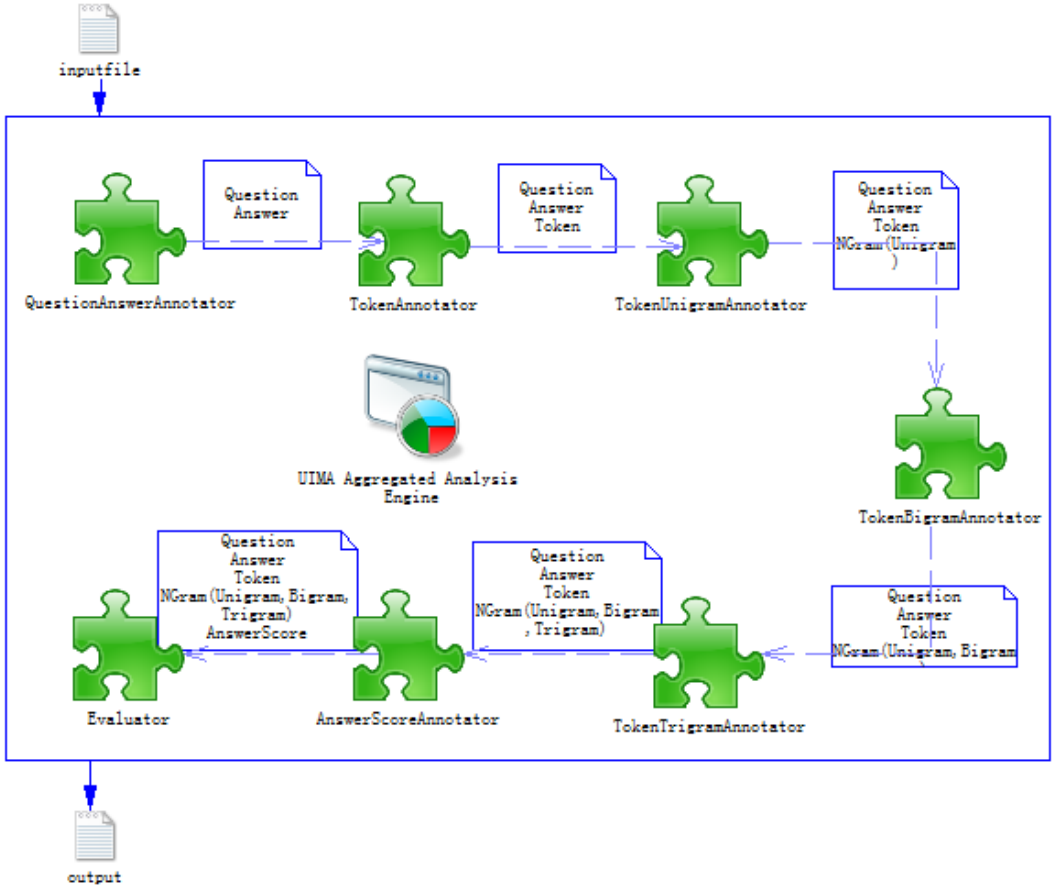

Figure 1: Aggregated Analysis Engine

AAE Implementation

Annotators

Totally, there are 7 Annotators in my AAE and I make use of hw2-kuol-aae.xml to aggregate them to achieve the final goal.

QuestionAnswerAnnotator:

This annotator is used for annotating Questions and Answers.

TokenAnnotator:

This annotator is used for annotating Tokens. It also makes use of the results of the QuestionAnswerAnnotator.

TokenUnigramAnnotator:

This is annotator makes use of the results of TokenAnnotator to annotate the unigrams.

TokenBigramAnnotator:

This annotator makes use of the results of QuestionAnswerAnnotator to annotate the bigrams.

TokenTrigramAnnotator:

This annotator makes use of the results of QuestionAnswerAnnotator to annotate the trigrams.

AnswerScoreAnnotator:

This annotator makes use of the results of QuestionAnswerAnnotator to annotate the AnswerScores.

Evaluator:

This annotator makes use of the results of QuestionAnswerAnnotator to output the final results and precision@N to standard output.

Implementation

When writing the code:

1. I take care to make full use of the results which are already there.
2. Also, when a piece of code might be used more than once, I write it as a function in order to reduce modification.
3. When a function won't be used outside the class, I declare it as private.
4. I declare most of the parameters in the descriptors and try to avoid using magic numbers.

When choosing the scoring method, I make use of the token overlap scoring method. The reason is that although NGram(N>1) is useful theoretically, when it comes to practice, it is not as powerful as unigram.