# 11-791 HW2 Report

Zhiyu Li

Sept. 22th, 2013

**Name: Zhiyu Li**
**Andrew ID: zhiyul**

## 1 Summarize

This report includes the analysis engines design specified in the homework 2 handout.

## 2 Pipeline Design

Similar to homework 1, this system's pipeline consists five components: **Test Element Annotator, Token Annotator, NGram Annotator, Answer Scorer and Evaluator.** Each component contains some inputs and outputs. In my design, I divide NGram Annotator into three sub-Annotators, namely UniGram Annotator, BiGram Annotator and TriGram Annotator. The input and output for different components are listed as below:

1. **Test Element Annotator:**
   Input: Text File
   Output: Question+, Answer+
2. **Token Annotator:**
   Input: Question+, Answer+
   Output: Token+
3. **UniGram Annotator:**
   Input: Question+, Answer+
   Output: NGram+
4. **BiGram Annotator:**
   Input: Question+, Answer+
   Output: NGram+

5. **TriGram Annotator:**

   Input: Question+, Answer+

   Output: NGram+

6. **Answer Scorer:**

   Input: Question+, Answer+

   Intermediate output: Tokens+ or NGrams+

   Output: AnswerScore+

7. **Evaluator:**

   Input: AnswerScore+

   Output: Accuracy (Print to the stdout so do not need to create a type for this)

The Evaluator component will be discussed in detail in the next section.

This order above is also the work flow of the processing pipeline. In each step, the corresponding annotator will assign its annotations the source and the confidence. Notice in the implementation, I did not use any NLP tools in component 2, 3, 4 and 5.

The reason all the three NGram sub-Annotators all produce the NGram-type instances instead of more granular NGram-type instances is in the NGram Overlap Scorer, all the NGram type will be considered as the same type regardless of the value of N.

For the Answer Scorer, depending on the scoring algorithms we are using, the inputs can be different (e.g. The N-Gram Overlap Scoring Algorithm will require the NGram+ from both Question+ and Answer+ as input, while the Token Overlap Scoring Algorithm will require Token+ from both Question+ and Answer+ as input). I implemented three different scorers in this component, namely *Token Overlap Scorer, NGram Overlap Scorer,* and *Golden Answer Scorer*. To use a specific scorer, just open the *hw2-zhiyul-aae.xml* and add the scorer you want to use into the component Engine Flow. Note that in the flow, the Scorer needs to be put after all the Annotator Descriptors, and before the Evaluator Descriptor. If you want to evaluate a specific scorer's accuracy, please put only that scorer before the Evaluator Descriptor. The scorer I use in this assignment is the *NGram Overlap Scorer*.

# 3 Evaluator: comparison between different scorer algorithms

I designed the evaluator component according to Alkesh Patel's email:

> *Evaluation component should just compare the gold standard and whatever output(0 or 1) your system gave. And print the accuracy. For example,*

*Q Who shot Lincoln?*
*A 1 1 Booth shot Lincoln*
*A 1 0 Booth killed Lincoln*
*A 0 0 Booth didn't kill Lincoln*

*where, first number is gold standard, second number is what was predicted by your system. According to this example, accuracy of the system will be, 2/3 i.e. 66.67%*

If we use "Token Overlap Scorer" or "N-Gram Overlap Scorer", the score for each answer could be a fraction (e.g. 0.25 for answer1 and 0.43 for answer2). To satisfy the evaluation method above, I used a threshold function to change the scores to {0,1}. For example, if the threshold is 0.33, then 0.25 will become 0 and 0.43 will become 1.

Below is the accuracy for NGram Overlap Scorer when threshold is 0.33:
1 1 Booth shot Lincoln.
0 1 Lincoln shot Booth.
1 0 Lincoln was shot by Booth.
0 0 Booth was shot by Lincoln.
1 1 Booth assassinated Lincoln.
0 1 Lincoln assassinated Booth.
1 0 Lincoln was assassinated by Booth.
0 0 Booth was assassinated by Lincoln.
Accuracy: 0.5
1 1 John loves Mary with all his heart.
1 0 Mary is dearly loved by John.
0 0 Mary doesn't love John.
0 0 John doesn't love Mary.
1 1 John loves Mary.
Accuracy: 0.8

Here is the accuracy for Token Overlap Scorer when threshold is 0.33:
1 1 Booth shot Lincoln.
0 1 Lincoln shot Booth.
1 1 Lincoln was shot by Booth.
0 1 Booth was shot by Lincoln.
1 1 Booth assassinated Lincoln.
0 1 Lincoln assassinated Booth.
1 1 Lincoln was assassinated by Booth.
0 1 Booth was assassinated by Lincoln.
Accuracy: 0.5
1 1 John loves Mary with all his heart.
1 1 Mary is dearly loved by John.
0 1 Mary doesn't love John.
0 1 John doesn't love Mary.
1 1 John loves Mary.
Accuracy: 0.6

| threshold | 0.16 | 0.20 | 0.25 | 0.33 | 0.5 |
|---|---|---|---|---|---|
| NGram | 0.45 | 0.45 | 0.65 | 0.65 | 0.55 |
| Token | 0.55 | 0.55 | 0.55 | 0.55 | 0.35 |

Table 1: Average accuracy for NGram Overlap Scorer and Token Overlap Scorer.

We can see under the threshold of 0.33, the average accuracy of NGram Scorer (0.65) is better than the average of Token Scorer (0.55).

We tuned the parameter threshold, and the results are shown in Table 1. When the threshold is small (<= 0.20), the Token Scorer has a better average accuracy than the NGram Scorer. When the threshold gets larger (>= 0.25), the NGram's average accuracy is better. Since the input data is too small, we could not draw a conclusion here to say which Scorer is better, but in this assignment, we stick with the NGram Scorer and set the threshold to 0.33.