

11-791 : Design and Engineering of Intelligent Information Systems

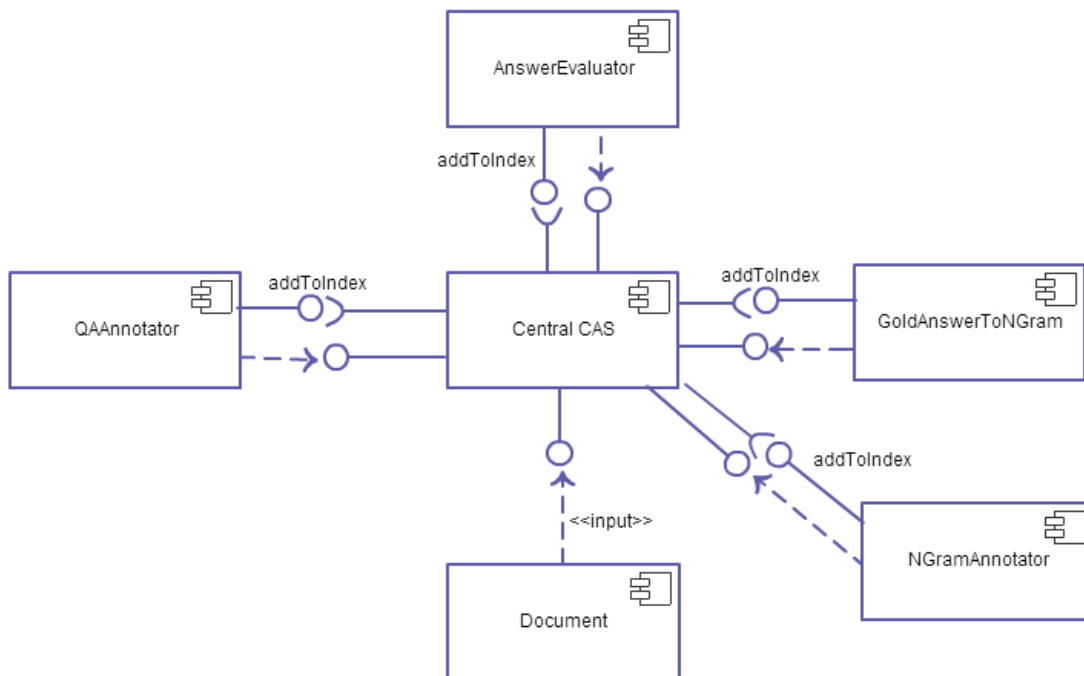
Homework 2

Name : Soumya Batra
Andrew ID : soumyab

Logical Architecture for the Problem Statement :

We have designed an aggregated analysis engine that takes a text document having one question and several answers as input and returns a ranked list of answers alongwith precision value at N correct answers. The aggregated analysis engine consists of a QAAnnotator, Gold Pipeline, NGram Annotator and an AnswerEvaluator.

UML Component Diagram: Let us have a look at the following UML component diagram to understand the interaction between various components at a high level:



Annotators Description: Let us look at the description of each annotator in detail:

1. **QAAnnotator** – This is the first annotator that gets executed in our Aggregated Analysis Engine. This takes the document text as input from the Central CAS and annotates all sentences as Questions/Answers and puts them back in CAS. Each sentence is differentiated from the other by the newline character.
2. **GoldAnswerToNGram** – This is the next annotator that takes Answers as input. It makes 1, 2 and 3-Grams of all answers that are correct (have attribute isCorrect = 1). It then puts all these NGrams back to the CAS.
3. **NGramAnnotator** – This next annotator takes the Question as an input and makes 1, 2 and 3-Grams of it. It adds these to the NGrams list obtained from Gold Answers from GoldAnswerToNGram annotator. With the combined list of NGrams, it annotates all the N-Grams of the answers (Gold or not). Also, it checks for 1, 2 and 3-gram overlap between the NGrams list and the answers. It obtains a score for an answer to be the correct answer as:

$$score = \left(\sum_{i=1}^3 \frac{i * No. of i - grams overlapped}{No. of i - grams in answer} \right) / 6$$

We multiply the N-gram by N, where N = 1, 2, 3 so as to give a higher weight to a 3-gram overlap than a 2-gram and 1-gram overlap and so on. Also, we divide the entire score figure by 6 in order to get a normalized term between [0, 1].

We utilize results from GoldAnswerToNGram annotator so as to identify more N-grams that may be present in a correct answer. By doing this, we get a perfect score value for all Gold Answers and a better estimate for all correct answers whose information about being/ not being Gold Answer is not known.

4. **AnswerEvaluator** – Finally, the AnswerEvaluator ranks all answers in decreasing order of score, outputs precision for each set of question and answers. Also, it prints average precision for all input sets.

Design Patterns:

1. **Low Coupling** – All annotators are independent of each other's functionalities. The input obtained by each can be gathered from any other source. Both input and output objects are put in the CAS by each annotator. This removes the dependency of every annotator on the other completely.

2. **High Cohesion** – All annotators take care of just the objects and functionalities they are concerned with and not interfere in the functioning of the others. QAAnnotator is concerned with annotating sentences into Questions and Answers, GoldAnswerToNGram is concerned with creating N-Grams of every Gold Answer, NGramAnnotator is concerned with creating N-Grams of all answers and questions and giving a score to each answer, and finally AnswerEvaluator is concerned with ranking all answers according to their scores and printing individual and average precisions.
3. **Sequential Flow** – There is a sequential flow from one annotator to the other. QAAnnotator is called first followed by GoldAnswerToNGram and NGramAnnotator, finally ending with AnswerEvaluator. No parallelism is involved anywhere and proper functionality of the entire system would be obtained only on running these annotators in this sequence.

Comparison with other methods:

1. **Running Gold Pipeline and N-Gram annotators independently** – We have run Gold Pipeline and N-Gram annotators in sequence instead of running them as independent units.

Reason for doing so:

Creating N-Grams from Gold Answers and then using these N-Grams along with N-Grams from questions, will give a better estimate of score for an answer whose correctness is not known. Also, for Gold Answers, perfect score of 1 will be obtained, which is the same behavior as of Gold Pipeline's.

2. **Giving weights to each N-Gram** – We have given weights to each 1-gram, 2-gram and 3-grams instead of having equal or weight = 1 to them.

Reason for doing so:

There is a higher probability of an answer having a matching 3-gram to be a correct answer than an answer having a matching 2-gram or 1-gram. Hence, 3-grams are given highest weight, 2-grams the next highest and 1-grams the lowest weight.

Interesting Discovery:

1. Calculating N-Gram overlap from Questions as well as Gold Answers give a better estimate of score for an answer whose belongingness to a Gold Answer list is unknown.
2. This method gives the same score to a Gold Answer as the Gold Pipeline method.
3. Giving more weight to an N-Gram of a higher degree gives a better estimate of an answer's score than giving equal weight to all N-Grams.