README V2.0 Nov 13th, 2019

** NOTE: Feel free to contact us for further clarifications. We will update the README file every now and then to make it clearer.

Subtasks:

1. Detection:
   - The system shall detect for each sentence, whether it is ambiguous or not, based on the anaphor in question.
   - For example, if the anaphor is preceded by three plausible antecedents then it might be ambiguous.
   - A sentence is **ambiguous** if it contains at least one ambiguous anaphor.

2. Disambiguation task:
   - The system shall suggest a possible resolution for anaphors that are found unambiguous.
   - For example, the closest noun phrase might be the most likely antecedent.

Training-set.csv

- The file contains a sentence id followed by the sentence text.
- The id represents the domain from which the sentence is taken and includes a unique number for that particular sentence; e.g., “railway#03” represents sentence number 3 from the “railway” domain.
- Each sentence contains at least one pronoun that is tagged with <referential>; e.g., “Once material has arrived, **<referential>it</referential>** must undergo several reviews, including virus checking, format compliance and anticipated content and file type.”
- NOTE: the tag doesn’t necessarily mean that the pronoun is ambiguous. The tags are simply added to (1) facilitate referring to the answer files (2) help the participants identifying the anaphors in question (3) have a unified view of the problem.
- If there is more than one pronoun, the tag will be distinguished with an **“id”** {a,b,c,...etc.} to enable referring to it in the answers files.

Answers-files.csv

- The answer files contain the same sentence id (as in the training set) and the corresponding solution according to the manual annotations.

- There are two answer-files, one for the ambiguity detection subtask and one for the disambiguation subtask.

1. **“detection_answers_file.csv”:**
    - Contains the results sentence-wise.
    - Each sentence is marked as ambiguous or unambiguous if it contains at least one referential ambiguity.
2. **“disambiguation_answers_file.csv”**:
    - Contains the results for each anaphor that is unambiguous according to the manual annotations.
    - For example: “**library#12-a**” refers to the anaphor with the id = ”a” in the sentence number 12 from the library domain.

Evaluators

- As mentioned in the description, we will evaluate using precision and recall.
- For **detection**, precision and recall will assess how well the system could identify (detect) the ambiguous sentences.
- For **disambiguation**, precision and recall will assess how well do the system suggestions match the resolutions in the gold standard (as in the answers file)
- The participating systems should produce resulting file(s) similar to the answers files published with the training set.
- The evaluator ~~will be provided later~~ is uploaded in “evaluator/” folder.
- The evaluator contains two components to evaluate *detection* and *disambiguation.*
- To use the evaluator, you need to:

1. Store the results of the system in a file similar to the one in the “dummy_[...].csv” files.
2. Download the jar file (evaluate.jar) somewhere
3. Make sure the gold & system files are in the same directory of the jar file
4. To use the evaluator for evaluating detection:

Java -jar evaluate.jar -detection [dummy_detection.csv] [gold_detection.csv]

5. Names of the files in brackets can be changed to the names you have, but they should be both “.CSV”
6. For evaluating disambiguation:

Java -jar evaluate.jar -disambiguation [dummy_resolution.csv] [gold_resolution.csv]