William Ibekwe (wibekwe)
Homework 2 report

EXTENDED TYPES

In the archtype sample given for hw2 there were types given to to help us start the homework. Because we could not edit them in anyway, I went ahead wrapped them using a list of types that extended the ones given. the are located in the subtype folder Here is the following List:

- **CASDocument(Annotation)**
  - Question(Question)
  - AnswerList(FSArray) of Answers(Answer)
- **SubQuestion(Question)**
  - QuestionString(String)
  - SentenceStructure(Annotation)
- **SubAnswer(AnswerScore) -**
  - IsScore(Boolean)
  - AnswerString(String)
  - SentenceStruncture(Annotation)
- **SentenceStructure(Annotation)** -
  - TokenList(FSArray) of Token(Token)
  - NGramList(FSArray) of NGramsSets(Annotation)
- **NGramSets(Annotation)**
  - NGramTokens(FSArray) of Tokens

ANNOTATIONS

The following are the description and implementation the annotators used in my project.

Document Annotator: The document annotator is used to seperate the question and answer list in the given cas document. I start out by by parsing each line and then parsing each line into it appropriate format so that it can be store in a certain type.

The input format is given with the following constraints:

Q <The question in string form>
A <Gold standard score> <The answer in string form>

The question string is stored in the Question data type for later annotations. The answers String are also stored in the the Answer data type for each Answer in the AnswerList. Once the parsing is done the both the question and the answer is stored back into the original CasDocument Data type and then indexed back into the JCas object for further use in the other Annotations. Also in each Answer the Gold standard Score is sent stored.

Token Annotator: The token annotator takes the string of both the question and answers within the answer list and parses them into individual words. removing any kind of punctuation or white space associated with them. The Question and answer List is

then retrieved from the cas object and so that their strings can be parsed and put inside the token list of each of their Sentence Structures.

NGram Annotator: The NGram annotator takes the sentence strings in both the question and answers of the list and then makes 1, 2, 3, and 4-gram lists with them. These lists are then put inside the ngramlist located in each question's and answer's Sentence Structure.

Answer Scoring Annotator: The Answer Scoring Annotator is the token list of each answer and compares it to the token list of the question and compares them using the Jaccard Similarities Algorithm. This algorithm is simply a matching algorithm that will separate the answers into 3 different categories: words that match, words that are in the answer string, and words that in the question string. Once this is done a certain answer will be given based of the cardinality of each set. Once the score is calculated it will be placed into each Answer's Score variable.

Evaluation Annotator: The Evaluation Annotator takes Score for each Answer and prints them out into the following form.

<"+/-" = the Golden Standard > <Score> <Sentence>

DESCRIPTORS

The following is a list of Descriptor typed used in the document analyzer:

TopLevelDescriptor: Aggregate of the following descriptors
DocumentDescriptor: Primitive
TokenDescriptor: Primitive
NGramDescriptor: Primitive
AnswerScoringDescriptor: Primitive
EvaluationDescriptor: Primitive