# Hw1-Kartikgo-report

Kartik Goyal
kartikgo@cs.cmu.edu

September 11, 2013

## 1 Design for Answer Ranking Task

The overall hierarchical structure of classes is organised as follows. Every class has a Meta field of type MetaData(Explained below):

Entity
- Question
  - Sentence
    - TokenList(FSList-Token)
      - Word
      - POS(part of speech tag)
    - UnigramList(FSList-Unigram)
      - token1
    - BigramList(FSList-Bigram)
      - token1
      - token2
    - TrigramList(FSList-Trigram)
      - token1
      - token2
      - token3
  - AnswerList
    - TokenList(FSList-Token)
      - Word
      - POS(part of speech tag)
    - UnigramList(FSList-Unigram)
      - token1
    - BigramList(FSList-Bigram)
      - token1
      - token2
    - TrigramList(FSList-Trigram)
      - token1
      - token2
      - token3

The classes implemented are:

- General.MetaData- This is a common attribute to all the classes. It accounts for the source(the component that generated this annotation), the confidence of it being an annotation, and start and end offsets.
  Features:
  - Start
  - End
  - Source
  - Confidence
- Model.Entity- This represents the whole input structure comprised of question and a list of answers.
  Features:
  - Question
  - AnswerList(FSList-Answer)
  - Meta
- Model.Question- This represents a question with a notable feature being Sentence
  Features:
  - Sentence
  - Meta
- Model.Answer- This represents a candidate answer with notable features being Sentence, Actual score(label), and predicted score.
  Features:
  - Sentence
  - True Score
  - Predicted Score
  - Meta
- Model.Sentence- This represents a sentence containing a list of tokens, a list of bigrams, a list of unigrams and a list of trigrams.
  Features:
  - TokenList(FSList-Token)
  - UniGrams(FSList-UniGram)
  - BiGrams(FSList-BiGram)
  - TriGrams(FSList-TriGram)
  - Meta
- Model.Token- This represents a token with it notable fields being word and POS(part of speech tag)
  Features:
  - Word
  - POS
  - Meta

- Model.UniGram- This represents a unigram with its notable field being Token.
  Features:
  - Token1
  - Meta
- Model.BiGram- This represents a bigram having two token features.
  Features:
  - Token1
  - Token2
  - Meta
- Model.TriGram- This represents a trigram having three token features
  Features:
  - Token1
  - Token2
  - Token3
  - Meta

This organization of classes is compliant with the task at hand as all the tokens, n-grams, questions and answers are represented as seperate annotations. The two fields in answer class- predicted score and actual score will help in evaluation of performance and ranking of answers.
Finally, this representation is modular and extensible. We can add features to any level in the hierarchy spanning from Entity to Token.