CMU 11791 fall 2013
Homework 2

Wei Zhang

Id: weizhan1

Note: please Do Not use JcasGen to re-generate the type system. Token.java and AnswerScore.java implemented java Comparable interface. Regenerate the type system will overwrite the "implements Comparable" at the class head.

Evaluation:

P@N for q001.txt and q002.txt are both 100%.

dependencies :

additional dependencies and resource that are used:

Memory Usage:

2G, specify in –Xmx, for loading the models for stanford NLP and wordnet.

Table1: additional packages

package	Resources
stanford-core-nlp 3.2.0	Src/main/resources/stanfordModels
jwnl 1.4_rc3	Src/main/resources/dict

Package Overview:

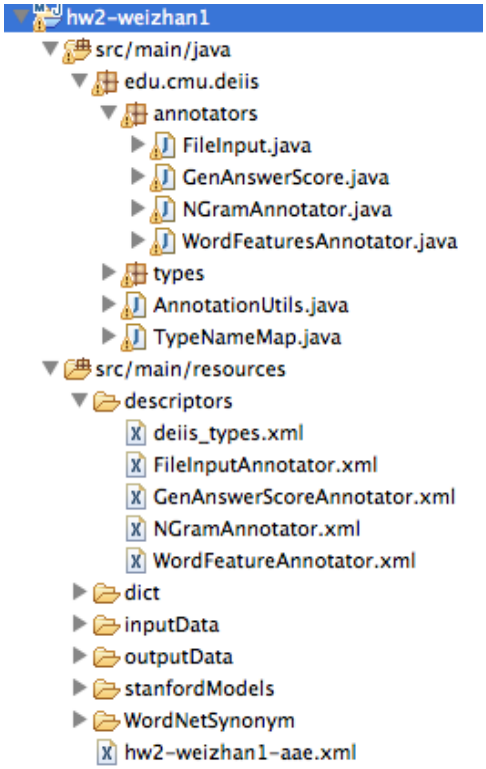

Fig.1 Project files overview

Additional helper class:
AnnotationUtils.java and TypeNameMap.java,
under src/main/java/edu.cmu.deiis package.

Type System Addition:

Table 2: type system additions

Type	**Addition**
Token	Added morph , part-of-speech, ner, and token.
Question, Answer	Added Dependencies for recording the dependency relations

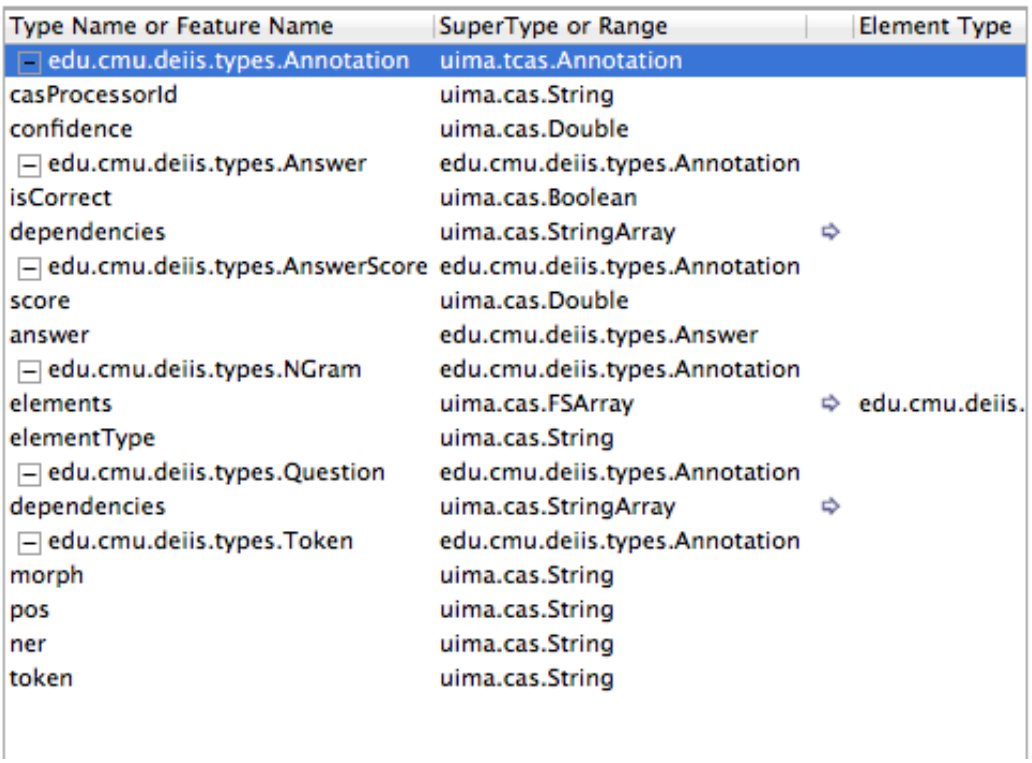

Fig 2: type system overview

Features:

1. Ngram feature(unigram, bigram, trigram)
2. bag of lemmas(lemmas for nouns), and
3. syntactic parsing features (subj-obj pair with additional polarity)to rank the answers.

See Next section for details.

Analysis Engines

Descriptors except the one for aggregate analysis engine, are located in src/main/resources/descriptors.

Engine java files are located in src/main/java/edu.cmu.deiis.annotators.

Correspondence:
Descriptors for the premitive analysis engines are as follows:

Table 3: annotator and description file correspondence

Annotator	Description file	parameters	Param value
FileInput.java	FileInputAnnotator.xml	N/A	N/A
WordFeatureAnnotator.java	WordFeatureAnnotator.xml	N/A	N/A
NGramAnnotator.java	NGramAnnotator.xml	n	Integer for specifying n for n-gram
GenAnswerScore.java	GenAnswerScoreAnnotator.xml	N/A	N/A

Capabilities :

Table 4: capabilities for AE

Description file	Capabilities
hw2-weizhan1-aae.xml	Answer,AnswerScore,Question,Ngram,Token
FileInputAnnotator.xml	Question, Answer
WordFeatureAnnotator.xml	Token
NGramAnnotator.xml	NGram
GenAnswerScoreAnnotator.xml	AnswerScore

AE Function Description:

Table 5: Function Description

Annotator	Methods used
WordFeatureAnnotator.java	Generate lemma, part-of-speech and syntactic parsing results. Lemma, pos are stored in the tokens, and parsing results are stored in the dependency field in Question or Answer.
NgramAnnotator.java	The ngrams are generated loosely, without attaching to the Question or Answer structures. However, the dependencies are attached to the Question or Answers.
GenAnswerScore.java	The generation of the Answer Scores, with feature generation and evaluation.

Feature

Ngram similarity feature: Generating the 1 gram and 2 gram and 3 grams for the Question and Answers, get the cosine similarity between each Question , Answer pair. The tokens uses the lemmas, instead of token original form.

Bag of nouns feature: Generating the lemmas of nouns for Question and Answer, get the cosine similarity between each QA pair.

Bag of dependencies: Each element in the bag of dependency is a agreement (binary value) between question dependency and answer dependency.

* agreement 1: obj+subj+polarity
* agreement 2: verb+polarity
* agreement 3: obj + verb + polarity
* aggrement 4: subj +verb + polarity

aggrement is true if question dependency is equal to answer dependency. Each string is reconstructed by following rules:

Map verb in wordnet to the same token, if they are in the same synset.
use token lemma for obj, subj, and verb.

So, the same dependency will be identical given the introduction of the knowledge base.

Several cases:

1.Booth shot lincoln -> booth+lincoln+false, booth+shot+false, shot+lincoln+false, shot+false
2. Booth is shot by lincoln -> lincoln+booth+shot, lincoln+shot+false, shot+booth+false, shot+false
3. Booth didn't shot lincoln -> booth+lincoln+true, booth+shot+true, shot+lincoln+true, shot+true.
4. Lincoln is shot by Booth. -> booth+lincoln+false, booth+shot+false, shot+lincoln+false, shot+false

Similarity between 1,2: 0.25.
Similarity betweetn 1,3: 0.0
Similarity betweetn 1,4: 1.0
polairt y denotes whether negative relation is contained in the dependencies.

Evaluation Method:

Evaluation is generated in the GenAnswerScore.java. There's some command line output in this class, too.

Evaluate with P@N, where N is the number of golden correct answers.
Method:

Ranking the AnswerScore first based on the score field (by implementing Comparable interface for AnswerScore class), then select the Top N answers to examine the number of correct answers.