11-791 HW 1 Report

Chenyan Xiong
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
cx@cs.cmu.edu

1 General Design Idea

In my design, the QA pipeline in homework 1 probability works as follows, step by step:

1. Input: a question and a set of answering sentences, each sentence is associated with a ground truth score, showing whether it is correct.
2. Tokenization: run a tokenizer for each sentence, annotate each token span of it.
3. NGram: given the tokenized sentences, annotate ngrams of them.
4. Prediction: takes the annotated sentences as input, predict the score based on a certain (for example, cosine similarity) scoring function.
5. Evaluation: given all sentences and their predicted score, evaluation the performance (I.E. MAP) based on given ground truth score.

From the pipeline one could see there are several level of information should be recorded during the process, and we design according the following type systems:

1. edu.cmu.cs.lti.cx.Token, inherited from uima.tcas.Annotation.
2. edu.cmu.cs.lti.cx.Ngram, inherited from uima.tcas.Annotation.
3. edu.cmu.cs.lti.cx.Sentence, inherited from uima.tcas.Annotation.
4. edu.cmu.cs.lti.cx.Question, inherited from edu.cmu.cs.lti.cx.Sentence.
5. edu.cmu.cs.lti.cx.Answer,inherited from edu.cmu.cs.lti.cx.Sentence.
6. edu.cmu.cs.lti.cx.Evaluator, inherited from uima.cas.TOP.

In the next section, we will discuss the features of each type system.

2 Detailed design of type systems

2.1 Token

The Token is used to record information for individual tokens in a sentence. It is inherited from uima.tcas.Annotation. Its features include:

1. postag: uima.cas.String, the postag of this token.
2. StemRes: uima.cas.String, the stemmed word for this token.
3. TermNum: uima.cas.Interger, the number of term of this token, mostly 1.
4. TF: uima.cas.Double, the term frequency of this term in current sentence.
5. IDF: uima.cas.Double, the invert document frequency of this term in corpus.

2.2 Ngram

Ngram is to record information for ngrams in a sentence. It is inherited from uima.tcas.Annotation. Features of it are:

1. NgramLen: uima.cas.Integer, the 'n' of this n-gram.
2. TF: same as it is in Token type.
3. IDF: same as it is in Token type.

2.3 Sentence

Sentence type is used to record information for question and answer. It is also inherited from uima.tcas.Annotation. As a result,the sentence is kind of an virtual type, most features are defined at its sub-classes: question and answer.

Question Question is a sub-class of Sentence:

1. QuestionType: uima.cas.String, the type of this question. I.E. what, why, how, etc.
2. ConfidentScore: uima.cas.Double, the confidence of the QA system in answering this question. If the score is too low, one could choose not to answer it.

Answer Answer is also a sub-class of Sentence:

1. PredictedScore: uima.cas.Double, the predicted score of this answer.
2. GroundTruthScore: uima.cas.Double, the ground truth score of this answer.

2.4 Evaluator

Evaluator is a class used for evaluation, it records input a ranking list of predicted scores and corresponding ground truth scores, and evaluation outputs like NDCG, MAP, Precision, etc. It is directly inherited from uima.cas.TOP, with features:

1. PredictedScoreArray: uima.cas.DoubleArray, records the scores of predicted ranking list.
2. GroundtruthScoreArray: uima.cas.DoubleArray, records the scores of corresponding ranking list.
3. NDCG: uima.cas.DoubleArray, evaluation results, NDCG at different position.
4. MAP: same as NDCG, but evaluation metric is MAP.
5. Precision: same as MAP, but evaluation metric is precision.