11791 HW2 Report

Anurag Kumar
Andrew id: alnu@andrew.cmu.edu

October 10, 2014

1 Aggregating Analysis Engines

The basic task is to use multiple analysis engine for Gene Tagging. The first step in this process is to read the input file through CAS collection reader. We have already been provided with a type system file. I added two types of my own. The first one "SentTS" is for input text. It contains two attributes one to store Sentence-Id and other for actual Sentence. The collection reader (NewCollectionReader.java) reads input file and splits sentence and sentence id. This is done line by line. The second type is "NamedTS" the super-type of which is the provided "Annotation" type (edu.cmu.deiis.types.Annotation). I added one more attribute to it to store the annotated named entity. The other used attributes namely confidence, begin and end offset are directly inherited from super type.

The problem requires us to implement multiple analysis engine and combine outputs from them. I implemented 3 analysis engine. Lingpipe's HMM based model, GENIA model and ABNER biomedical named entity recognizer. ABNER is a Conditional Random Field based biomedical named entity recognizer. Individual test on these models showed that performance on geneia model was very slow and hence I decided to remove the GENIA based analysis engine from system. So finally my aggregate engine analysis consists of only HMM based model and ABNER model.

I used the lingpipe in following way. Lingpipe's Confidence Chunker can return any desired number N-best chunks. I set N = 10. I choose only a subset of the returned named entities using the confidence score. If the confidence is above some threshold then it is added to CAS otherwise not.

By some experimentation on the given sample.in I fixed the threshold as 0.6. For ABNER, I simply fix a rule based on the length of the returned named entity. If the length is more than 8 I include it otherwise I don't. The outputs of two models are stored in CAS and finally the union is done in CAS consumer(CASConsumer.java) and written to the output file. The results on the provided sample.in is

- Precision: 0.599459563791
- Recall: 0.850205310704
- F1 Score: 0.703146932307

Some of the important aspects of the implementation are

- Line by Line reading input file
- The sentence id and text are separated during the collection reader process.
- Machine Learning Techniques
  - Lingpipe (HMM based model) based pre-trained model is first analysis engine. 10 best chunks are obtained and a named entity is considered if only the confidence is above 0.6
  - ABNER (CRF based model) is used as second analysis engine. Here the constraint is put on length, if it is more than 8 it is considered otherwise.
- The aggregation of analysis engine is done by union of the two analysis engine