11-791: Design and Engineering of Intelligent Information System

Homework 2

Name: Yulong Pei
Andrew ID: yulongp

1 Architecture

In this homework, the task is also to implement a named entity recognition system. Similar to Homework 1, the process of this system is shown in Figure 1. The difference is in annotating documents, multiple annotators are used in this homework.

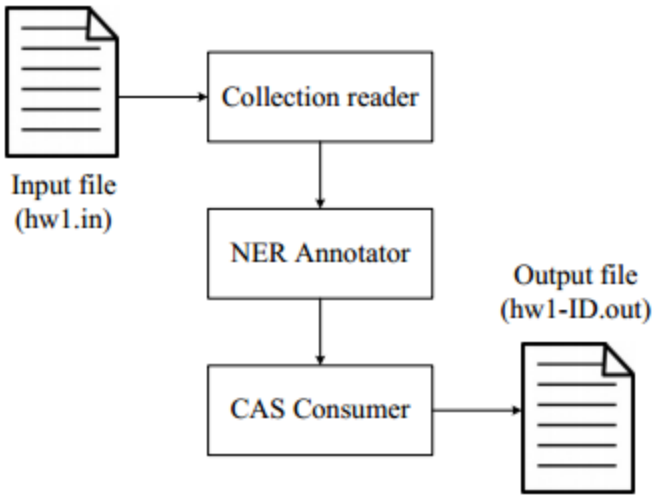

Figure 1: The process if a typical NER system.

1.1 Type System

A standard type system is provided by TAs named deiis_type.xml and in this file, several types are defined including Annotation, Question, Answer and etc. However, these types are not relevant to the NER task. Therefore, based on Annotation, I define the type GeneEntity, which contains a string, a begin index and an end index. Furthurmore, in order to distinguish different annotators, based on the type GeneEntity, two types are inherited, i.e., GeneEntitybyPOS and

GeneEntitybyLingPipe. These two types have an extra feature, i.e., the casProcessId which is used to represent the specific annotator.

Besides, similar to last homework, Sentence type is also defined which is also based on Annotation type. Sentence contains a string and a sentence id.

1.2 Collection Reader

The collection reader uses a BufferedReader to read the input file. Each time on line, i.e., on sentence is stored into JCas with the sentence id and corresponding text. The parameter for collection reader to get the input file path is set to be "hw2.in".

1.3 Aggregate Analysis Engine

In this compennet, two annotators have been applied and the structure for these annotators are shown in Figure 2.

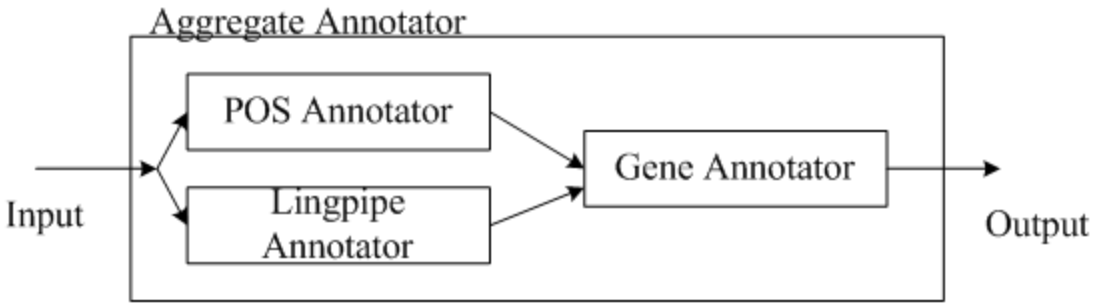

Figure 2: The struture for individual annotators and aggregate annotator.

1.3.1 POS Annotator

This annotator is the same in Homework 1 so the details are ignored here. It is worth mentioning that the confidence for all the entities extracted by POS annotator is set to be 1.

1.3.2 Lingpipe Annotator

The Lingpipe Annotator is used the Lingpipe toolkit to train a offline NER model and then in the code, this mode is used to create a NER class to recognize the

named entity. The training process and the usage refer to the Named Entity tutorial[1]. And the data set used to train a NER model is the GeneTag named entity data[2].

In particular, I choose the N-Best Named Entity Chunking in Lingpipe for this homework and therefore multiple possible NE could be extracted with the confidence value.

1.3.3 Aggregate Annotator

To combine these two annotators introduced above, a simple strategy is used here which is if an entity is extracted by both annotators, this entity will be regarded as the gene entity. Since Lingpipe model uses more complicated method and trains on a gene data, more importance should be placed on this confidence value. Thus, after calculating this combination score, the entity extracted by Lingpipe annotation will be more likely to be the gene entity.

1.4 Cas Consumer

The cas consumer deal with output. o has a output file configuration parameter, which is set to "hw2-yulongp.out".

2 Performance

The performance is eveluated on the sample.in and sample.out file.
Precision: 0.801609517145
Recall: 0.50172460991
F1 Score: 0.617166717177

3 Problem

In this homework, an issue remains to be thought. In fact, I trained another model using Abner and the performance seems better than Lingpipe. However, the trained model is stored in a binary file and UIMA (in this scenario from my personal understanding) can only read into the file using stream operation. So I have no idea about how to load this model to improve the performance.

---

[1] http://alias-i.com/lingpipe/demos/tutorial/ne/read-me.html
[2] ftp://ftp.ncbi.nlm.nih.gov/pub/lsmith/MedTag/medtag.tar.gz