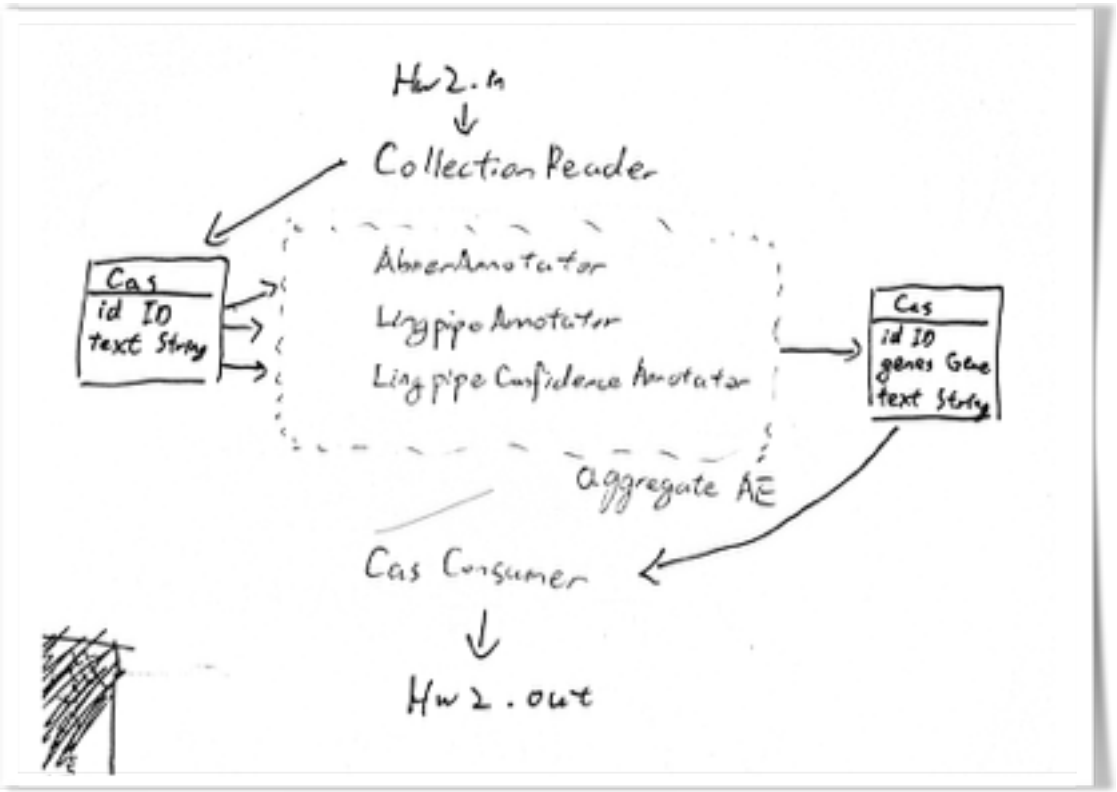
11-693 HW2 Report

Sun Xing, Andrew ID: xings

October 10, 2014



1. Introduction

In this homework we need to use an aggregate analysis engine to annotate the corpus instead of using only one AE to further improve the performance of our AE. The challenge is that in HW1, I used a model in Lingpipe that overfits the sample testing data thus we cannot get reliable F-score using the testing data given. The problem still exist since I cannot find other Lingpipe model that do not overfit. I used two other annotator for hw2 in addition to the 1st best name chunker in LingPipe that I used in hw1, namely the confidence chunker in LingPipe and abner. I evaluate the annotation produced by the 3 AEs and then calculate the words that is likely enough to be a gene mention. Since I do not have unbiased testing data, I did not try to train a weight for each of the annotators for evaluation. Instead, I set arbitrary weights during evaluation.

2. Architecture

To have a brief idea of the system that I built, please refer to the picture on the cover page. The following part presents the detail of my project.

2.1. Type System

The type system is specified in resources/deiis_types.xml. The types that are used in this project are listed below.

Type	features	description
Annotation	casProcessorId	records the analysis engine that produced the annotation
	confidence	records the confidence of the annotation
Gene	start	records the starting point for the gene in the according text
	end	records the ending point for the gene in the according text
	name	records the gene name
ID	id	records the ID of each jCas, in this case, each jCas is a line of text

2.2. Collection Reader

The collection reader is implemented in src/main/java/edu.cmu.xings.cpe/CollectionReader.java. The descriptor is in resources/collectionReaderDescriptor.xml This collection reader reads the input data (hw2.in) generate a JCas for each line in the text. It also annotate the ID of each JCas, which is the gene ID in the output. In this way we do not have to worry about ID of each gene during the annotation process.

2.3. Analysis Engine

The descriptor of the annotation engines located in src/main/resources/hw2-xings-aae.xml. In this project we used an aggregate AE combining 3 analysis engines: abner, 1st best name chunker in LingPipe, and confidence chunker in LingPipe. (details discussed in part 3.) These annotator add annotation Gene to the JCASs produced by collection reader. In the Gene annotations, we have the offset (starting location and ending location) of the potential gene mention, the annotator ID of the primary AE that generated it, the name of the gene, and the confidence calculated by the annotator. The annotations that are generated by the aggregate AE are then passed to the CAS consumer.

2.4 CAS Consumer

The description of the CAS consumer is located in src/main/resources/casConsumerDescriptor.xml. This CAS consumer go through all the JCAS to collect and process all the annotations produced previously. For each JCAS (represents a line in the input data), it gets the ID (generated in collection reader) and Gene (generated in the aggregate AE). For each gene, it calculates the overall confidence by summing up (with arbitrated weights) the confidences that each of the annotator predicts and decide if it is a gene mention. (details discussed in part 3.) If so, record it together with the ID of the JCAS. Finally it outputs all the gene mentions in the hw2-xings.out.

3. Algorithm

3.1 Analysis Engines

The analysis engines that are used in this project are listed below

abner	abner is developed for NER in biology texts. Its gene recognizer is trained on NLPBA corpora and BioCreative model using linear-chain CRF. The reported F-1 Scores are 65.1 and 69.9 respectively.
LingPipe	LingPipe Provides models for gene-tag and genomes. It has several different chunkers. In this project I used the confidence chunker and 1st best name chunker for the two different models. The two AEs are described below:
LingPipe - gene-tag	For gene-tag, it is supposed to find out the genes in general which is quite tough. I used confidence chunker to list the 5 potential genes with highest confidence. It will be judged in the evaluation part in CAS consumer.
LingPipe - genome	The names retrieved in this model is relatively straightforward without many different possibilities, so in this AE I used first best name chunker to annotate gene.

3.2 Combining the Annotations to Find Gene Mentions

To combine the annotations produced by all the annotators, I take the weighted average of the confidence of the annotators. For 1st best name chunker and abner, the confidence is 1, while for confidence chunker the confidence is the confidence calculated by the chunker. The formula of overall confidence is given below:

Overall.Confidence = abner.Confidence*w1 + LingPipeGenome.Confidence*w2 + LingPipeGeneTag.Confidence * w3

If the overall confidence is higher than the threshold, we say that it is a gene mention. This can be done using a 4th annotator but since it is only simple algebra, I implemented it directly in the CAS consumer.

Ideally, the weights and the threshold can be optimized using machine learning tools such as logistic regression. However, the overfitting issue in the given sample data makes it pointless to do this since one of our primary annotator is overfitted. Therefore, in this homework, the weights are arbitrary. Still I tried to make sense out of the weights and the threshold.

w1 = 0.25, w2 = 0.25, w3 = 0.5

Threshold = 0.3

Since the LingPipe gene-tag model's performance is the best among these 3 annotators and it has a confidence value to indicate the likelihood, we give it a higher weight of 0.5. And the other two are assisting the LingPipe gene-tag AE, So their weight are both 0.25. The threshold here is set to 0.4. In this way the two assisting annotator can never generate gene mentions by themselves, while the LingPipe gene-tag can as long as the confidence > 0.8. By using this reasonable parameter set, hopefully we can achieve higher f-score on the unknown testing data.

4. Performance Analysis

Although the sample.in and sample.out are not reliable in estimating the true f-score, we list the scores for reference. I tried different set of weight and thresholds and this result is not the highest among all the set-up. However we be careful to avoid w3 to be too large thus to reduce the risk of overfitting.

Precision	0.724241346007
Recall	0.780071174377
F-Score	0.751120248827