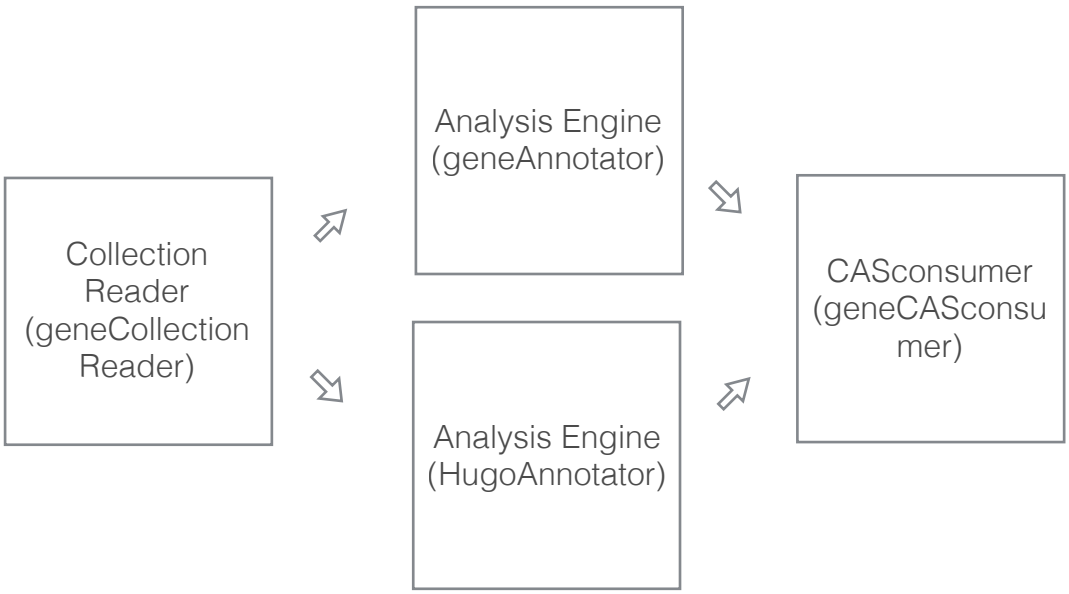
Report for 11791 HW2

Name: Chao-Hung Chen
Andrew ID: chaohunc

System Overview



Design and Algorithm aspects of system

In the previous homework, I used Lingpipe HMM models to build the analysis engine. However, even though the performance of Lingpipe model is good, Lingpipe model might overfit the data. To make my system more robust, I think it's important to have more models to improve the performance. However, after trying some models from other sources, I found that the performance is not pretty good. Therefore, I believed maybe it's important to have more training data or even an thesaurus, which included all gene names to avoid for matching unseen patterns (gene names). I then try to search the online source and find HUGO, which could be used for building a thesaurus for matching geneName. Although the conditions of matching gene names in HUGO, which had well defined information for each gene, is very strict, it definitely could increase the precision of discovering gene names. On the other hands, I also get insights from observing mislabel data. I found that it's very important to see whether the candidate include a keyword

(highly gene related). If this is the case, we should also consider the candidate as gene name.

Type System

GeneObj

Type or Feature	SuperType	
geneObj (Type)	uima.tcas.Annotation	
ID (Feature)	uima.cas.String	used to record ID
geneName (Feature)	uima.cas.String	recognized gene name from NER
posStart (Feature)	uima.cas.Integer	record the start position of gene in the sentence
posEnd (Feature)	uima.cas.Integer	record the end position of gene in the sentences
confidence (Feature)	uima.cas.Double	record the confidence scores of gene from NER

HugoGeneObj

Type or Feature	SuperType	
HugoGeneObj (Type)	edu.cmu.deiis.types.Annotation (Inheritance from given Annotation)	
casProcessorId (Feature)	uima.cas.String	used to record ID
geneName (Feature)	uima.cas.String	recognized gene name from NER
confidence (Feature)	uima.cas.Double	record the confidence scores of gene from NER

Collection Reader (geneCollectionReader)

As the collection reader, geneCollectionReader first need to read the file. Since the input file in this homework contains sentences, which contain one to several gene in the sentence, I then decided to read the file line-by-line and then combined those lines to put them as the input of geneAnnotator.

Analysis Engine (geneAnnotator)

geneAnnotators played the important role as the analysis engine in the system. I tried to use the model, provided by lingpipe, as core of analysis engine. It used HMM (Hidden Markov Model) to train a model of name entity recognition. For each sentence in the CAS, we put it in the model and check if there exists any possible gene. Based on the model, we could get the possible gene and its confidence of score. Then, we put the all of possible genes into the CAS.

Analysis Engine (HugoGeneAnnotator)

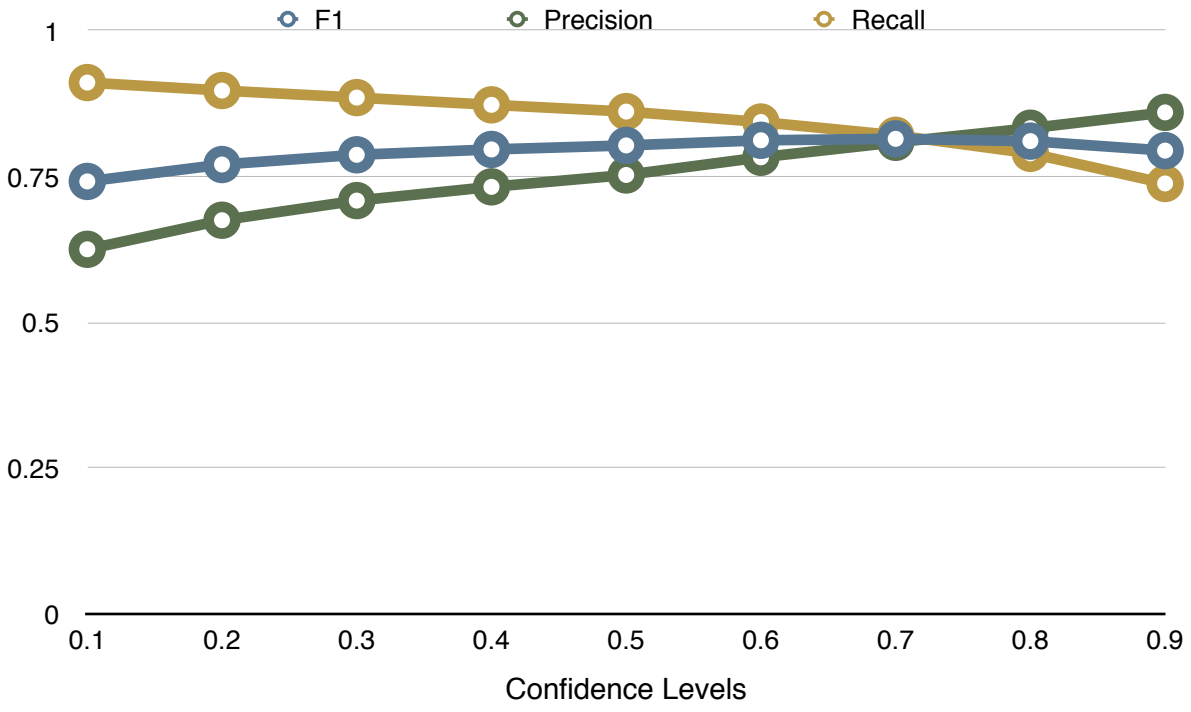
The key concept of HugoGeneAnnotator is to use gene thesaurus to match the keyword in the given sentence. The gene thesaurus is provided by HUGO Gene Nomenclature Committee (HGNC) ,which is the worldwide authority that assigns standardized nomenclature to human genes. It contains difference kinds of resource, including approved name of gene, previous approved name of gene, approved symbol of gene, previous approved symbol of gene, and synonyms.

In HugoGeneAnnotator, I tried to integrate all resources to maximize the utilization by using mainly two methods, “exact matching” and “partial matching”. For exact matching, it needs to match the sentence with the approved name of gene and previous approved name of gene. If they are matched, it means that those candidate gene name are very likely potential genes, which will be sent as CAS with high confidence scores. For partial matching, we only match each single word in the sentence with a gene dictionary. This gene dictionary is made up with approved symbol of gene, previous approved symbol of gene, and also the gene-related terms, which were extracted from approved name of gene, previous approved name of gene. The “partial matching” is also important for recognizing gene names, however, since it could only match partially, it should have lower confidence scores than “exact matching”. These “partial matching” potential gene name will also be sent to the CAS consumer.

CASconsumer (geneCASconsumer)

geneCASconsumer was used to integrate two results of CAS from two Analysis Engines and output the final result. To integrate the result , we first use CAS from geneAnnotator. From geneAnnotator, we could obtain basic confidence scores from CAS, and then match these candidate gene names with the result of CAS from HugoGeneAnnotator. If they match successfully, the confidence scores will be sum. If the sum of scores larger than threshold, I then extracted the potential gene from CAS and put them in the output sentences.

Experiment of parameters for confidence levels



Confidence Levels	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
F1	0.7419	0.7709	0.7876	0.7968	0.8037	0.8125	0.8150	0.8111	0.7948
Precision	0.6254	0.6753	0.7090	0.7326	0.7530	0.7833	0.8090	0.8333	0.8604
Recall	0.9116	0.8979	0.8858	0.8734	0.8618	0.8439	0.8211	0.7901	0.7384

The experiment result shows that when we set the parameter at lower confidence level, we will get high recall and low precision, and vice versa. Therefore, we try to use F1 measurement to pick our parameter, which is 0.7.