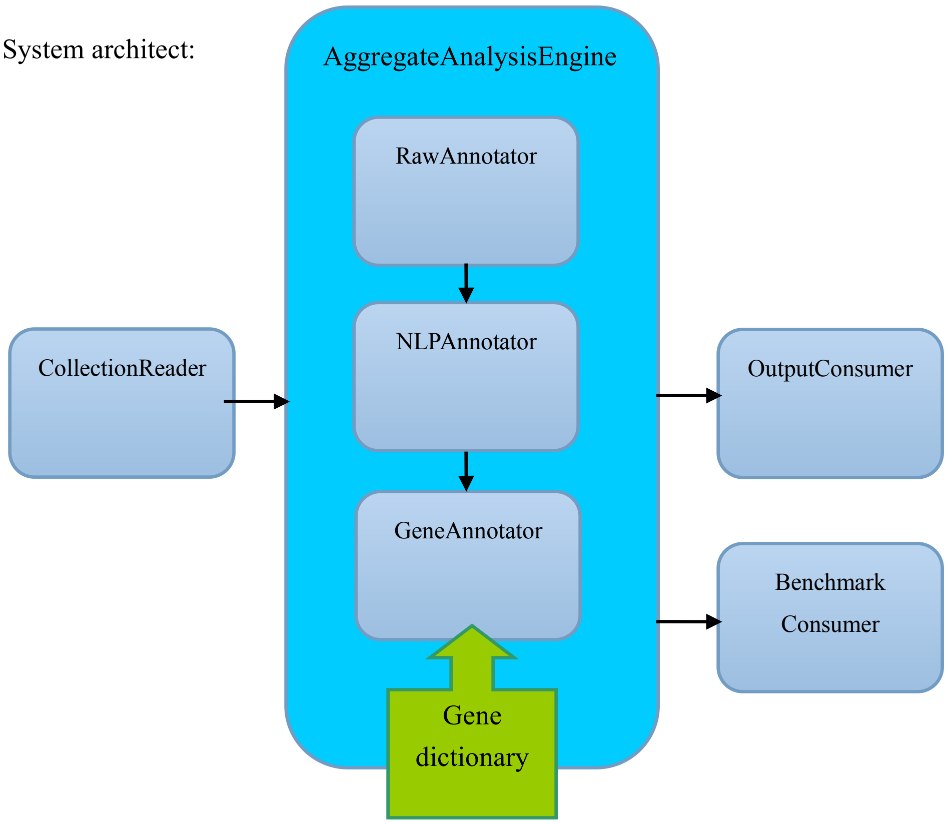
Report for Homework1: Xingyu Yan(xingyuy@andrew.cmu.edu)

1) System architect:



A) CollectionReader: reading file line by line and construct each line into a CAS.

B) AggregateAnalysisEngine: processing each CAS and generate gene name at the end, it incluses:

a) Raw Annotator: using regex to split each CAS content into 2 parts: it's key, and it's real content;
b) NLP Annotator: processing the real content of each CAS, annotate the parts which is noun phrase;
c) Using real content and NLP annotation, annotate gene name. As it's a multi-comparasion task, using trie tree as the dictionary container.

C) OutputConsumer: generate the output file of gene annotation in the required form.

D) BenchmarkConsumer: run benchmark using given answer file. Generate the benchmark and print it out in System.out.

E) Gene dictionary: A existed database of gene, contained many other informations. Downloaded from ftp://ftp.ncbi.nih.gov/gene/DATA/GENE_INFO/All_Data.gene_info.gz.

2) Outsource Using:

a) ftp://ftp.ncbi.nih.gov/gene/DATA/GENE_INFO/All_Data.gene_info.gz Gene database;
b) Stanford NLP tool

3) Pre-processed data: The gene database data:

This data is originally got 1.7G after unzipped. Soon I generated a name file but still rejected by git-hub due to it's "not over 100M" rule. Then I combined those genes using general suffix flag and finally zipped it to 8M.

4) Interesting things

Besides the gene database zipping problem, the containing of the data is trouble some as well. I used Hashset at the beginning, but after zipped the data, it'll no longer usable due to all the gene words became patterns. I choose trie tree as the saving structure and redesigned the matching process.

I once thinks a good gene dictionary is enough, then I found out many gene I extract is 'with' or 'was', and they are gene name. So I had to merged the NLP to eliminate those cases.