Your Name: Junjia He

Your Andrew ID: junjiah

Homework 2 Report

1. Statement of Assurance

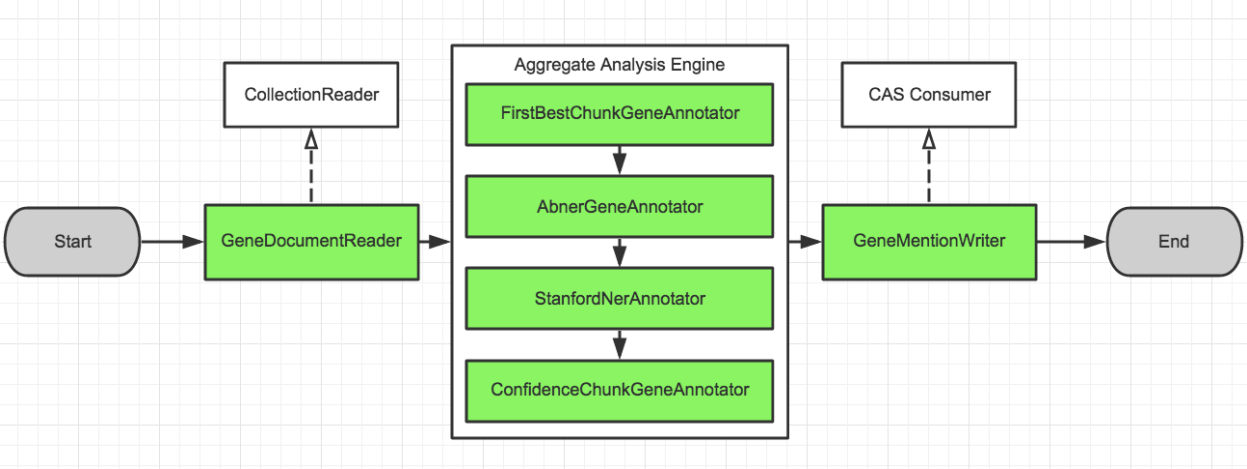
You must certify that all of the material that you submit is original work that was done only by you. If your report does not have this statement, it will not be graded.

I hereby certify that all of the material that I submit is original work that was done only by me.

2. Architecture

The architecture is similar to the one in homework 1. The major modification is on the annotator and the CAS consumer, which will be elaborated in the following sections.

2.1 Pipeline



In my pipeline, the first part is the same as before: *GeneDocumentReader* simply reads the source document line by line and splits the line to ID part and text part, then adds them to the CAS.

Then the CAS flows to the aggregate analysis engine (AAE). My AAE has 4 separate annotators: *FirstBestChunkGeneAnnotator*, *AbnerGeneAnnotator*, *StanfordNerAnnotator* and *ConfidenceChunkGeneAnnotator*.

FirstBestChunkGeneAnnotator is a named entity chunker from LingPipe library trained on GENIA. *AbnerGeneAnnotator* is based on ABNER, a biomedical named entity recognizer which in this homework is trained on BioCreative. *StanfordNerAnnotator* is adapted from the famous Stanford NER toolbox, and used the classical 3class model for English. At last, *ConfidenceChunkGeneAnnotator* is also from LingPipe library but trained on different corpus, GeneTAG.

In homework 1, *GeneMentionWriter* is a CAS consumer which writes the extracted Annotations (genes) to a file conforming to the format specified in the writeup. It reads the document ID and calculates the offset excluding white spaces. However in homework 2 *GeneMentionWriter* is also responsible to 'merge' the scores acquired from different annotators.

Note that all parameters (input/output file location, model file location, etc.) are all passed to the components by parameter setting in the corresponding descriptor instead of hard-wiring in the source codes. In addition, all model loading are done only in initialization. For further information please refer to the Javadoc or the code.

2.1 Annotators

- *FirstBestChunkGeneAnnotator:* This annotator uses first-best chunking to tag the genes, and I assigned confidence 1.0 for each tag. I used the model `ne-en-bio-genia.TokenShapeChunker` from LingPipe.
- *AbnerGeneAnnotator:* ABNER is an open source tool for automatically tagging genes and proteins, and in my annotator it's trained on **BioCreative** corpus. Since it does tokenization before tagging, some recognized entities cannot be matched to the original text, which I simply ignored. The confidence is also 1.0 since no probability is given.
- *StanfordNerAnnotator:* Stanford NER is not quit suitable for our task since the available models are basically all trained on non-biomedical corpus. Nonetheless, I added it to the aggregate analysis engine trying to find some entities which may not be recognized by other biomedical-related NERs. The confidence is also 1.
- *ConfidenceChunkGeneAnnotator:* This is also a chunking NER from LingPipe, but it returns chunks in order of confidence, therefore I directly recorded the confidence to the annotation. Note that the model I used is `ne-en-bio-genetag.HmmChunker` .

2.3 Confidence Aggregation

Scores are aggregated in the CAS consumer, namely *GeneMentionWriter*. I simply used weighed sum to calculate the final score and used a threshold to reject poor annotations. Note that all the weights and the threshold are heuristics based on my experiments.

Specifically, in *GeneMentionWriter* I iterated all annotations and used a HashMap to store the scores (note that I created a wrapper class for annotation such that annotations are equal if and only if their offsets are the same). For each annotation from different annotator, I multiplied its score with a pre-defined weight separately and sum them up. Finally I only wrote the annotations with scores higher than a pre-defined threshold to the output.

3. Evaluation

	`hw2-junjiah-aae`
precision	0.790408
recall	0.623542
F1 score	0.697129
Total Processing Time	401.35 s

The evaluation is conducted on the sample output. Since I heard (from Piazza) that the GeneTAG model from LingPipe may be overfitting to the sample input, I used models trained on other models to balance the results.

Annotator	**Running Time**
FirstBestChunkGeneAnnotator	267s
AbnerGeneAnnotator	79s
StanfordNerAnnotator	42s
ConfidenceChunkGeneAnnotator	9s