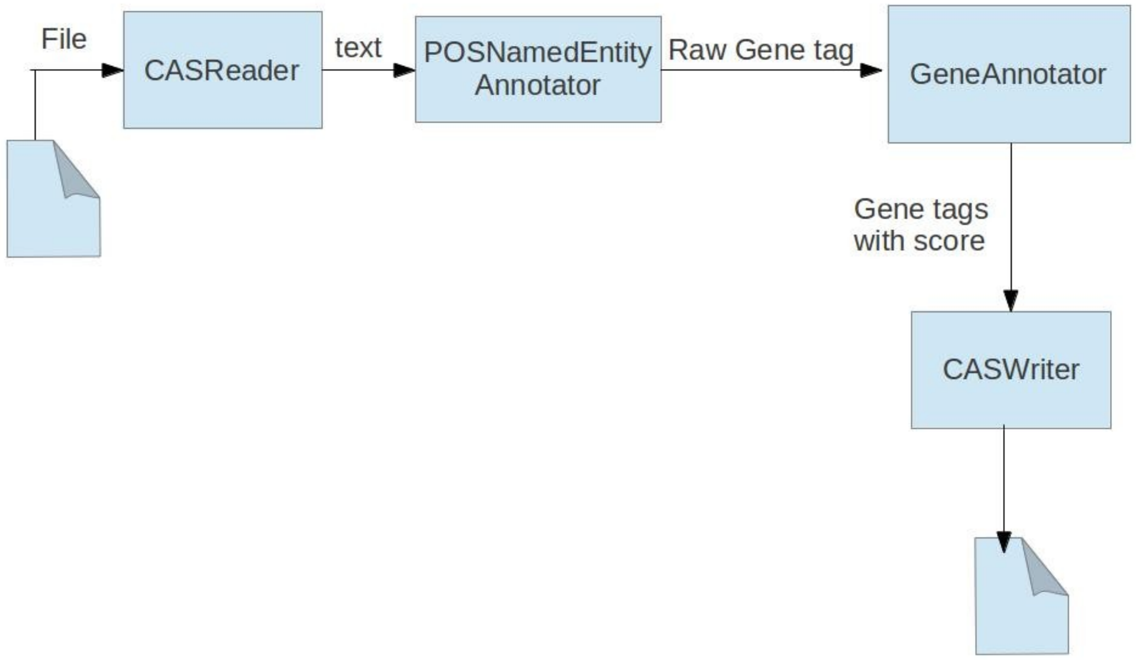
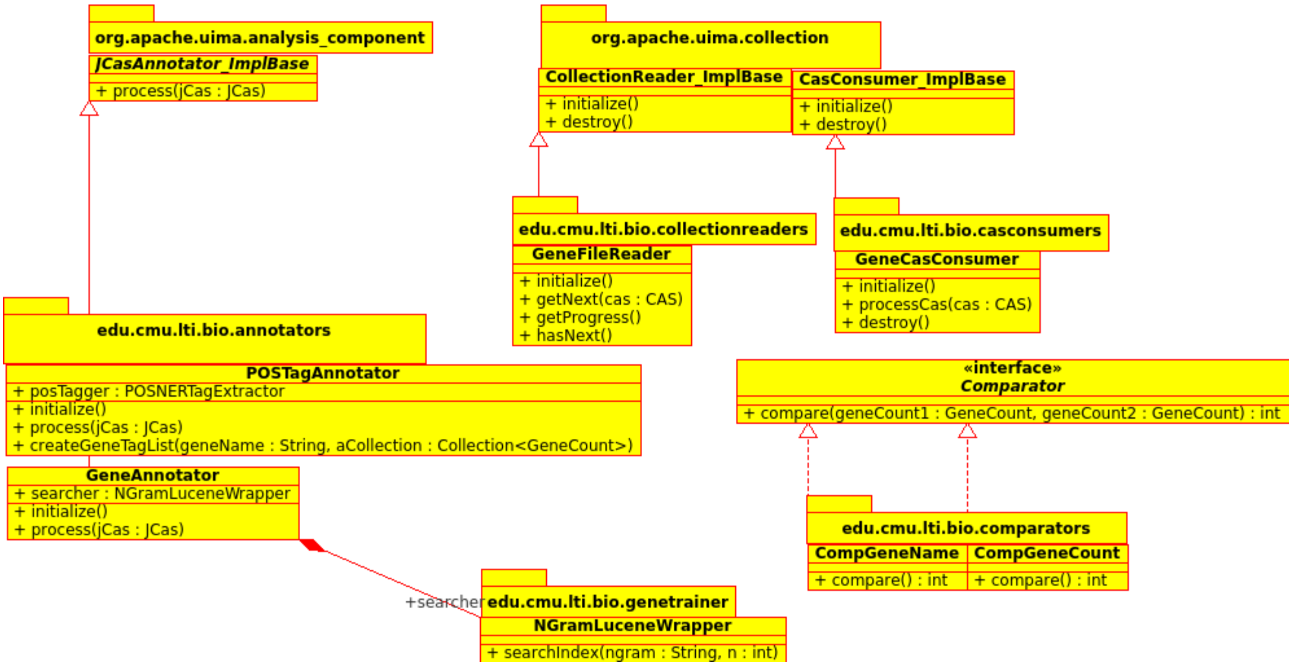
Task 3: Implementing A Named Entity Recognizer

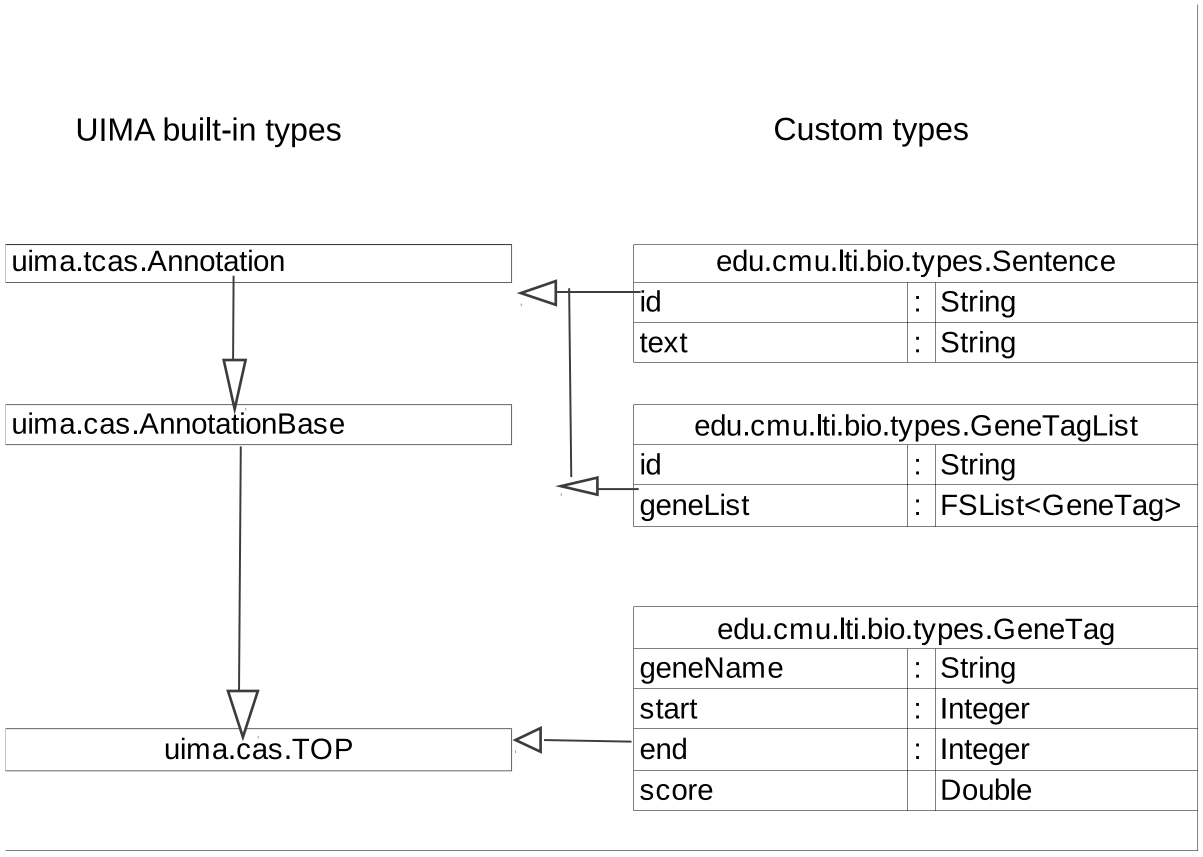
1. High-level architecture



2. UML Class Diagram


org.apache.uima.analysis_component
JCasAnnotator_ImplBase
+ process(jCas : JCas)
org.apache.uima.collection
CollectionReader_ImplBase
+ initialize()
+ destroy()
CasConsumer_ImplBase
+ initialize()
+ destroy()
edu.cmu.lti.bio.collectionreaders
GeneFileReader
+ initialize()
+ getNext(cas : CAS)
+ getProgress()
+ hasNext()
edu.cmu.lti.bio.casconsumers
GeneCasConsumer
+ initialize()
+ processCas(cas : CAS)
+ destroy()
edu.cmu.lti.bio.annotators
POSTagAnnotator
+ posTagger : POSNERTagExtractor
+ initialize()
+ process(jCas : JCas)
+ createGeneTagList(geneName : String, aCollection : Collection<GeneCount>)
GeneAnnotator
+ searcher : NGramLuceneWrapper
+ initialize()
+ process(jCas : JCas)
«interface»
Comparator
+ compare(geneCount1 : GeneCount, geneCount2 : GeneCount) : int
edu.cmu.lti.bio.comparators
CompGeneName
+ compare() : int
CompGeneCount
+ compare() : int
+searcher
edu.cmu.lti.bio.genetrainer
NGramLuceneWrapper
+ searchIndex(ngram : String, n : int)

3. Type System



edu.cmu.lti.bio.types.Sentence		
id	:	String
text	:	String

edu.cmu.lti.bio.types.GeneTagList		
id	:	String
geneList	:	FSList<GeneTag>

edu.cmu.lti.bio.types.GeneTag		
geneName	:	String
start	:	Integer
end	:	Integer
score		Double

4. NLP Tools Used

* Stanford-Core-NLP toolkit was used for following tasks in the UIMA pipeline:

- Tokenizer
- Part-Of-Speech Tagger
- Name Entity Recognizer

* Lucene was used to index N-Grams extracted from Corpus.

5. External Resource

GENETAG Training data was used for creating dictionary.
Genomics TREC -2006 data was used for extracting 5-Grams.

6. N-Gram Model

Genomics Track data was pulled from BIO-OAQA repository and upto 5-gram analysis was done. All N-Grams were indexed using Lucene to create index for faster search. Following steps were followed for determining whether given noun-phrase qualify for gene name:

- Extract noun-phrases using Standford-CoreNLP toolkit
- Search each noun-phrase in Lucene Idex
- Apply threshold on relevance score returned by Lucene for a noun-phrase in order to

```
        select the best gene name based on highest relevance score
```

7. Future Work

The threshold used for filtering best gene names is by gutt feeling. In future, I would look for detailed evaluation mechanism to figure out where system performed good and where it should have perform better. Would also look for better noun-phrase extraction strategy.