1 Collection Processing Engine

The Collection Processing Engine as described by CpeDescriptor.xml is responsible for reading the gene sentence data from the specified input file, performing annotations on it, and then outputting the sentences in their annotated form into the specified output file. This collection processing engine processes the input file sentence-by-sentence, to ease the annotation process for each analysis engine. Instead of having each analysis engine responsible for processing the entire file, sentences are streamed out one-by-one, to simplify the offset calculations of the gene mentions. The Collection Processing Engine is comprised of three parts:

1. SentenceCollectionReader:

   The SentenceCollectionReader is a simple collection reader that reads from an input file (`hw1.in`), and outputs its content, sentence-by-sentence into the CAS.

2. AggregateAnalysisEngine

   The AggregateAnalysisEngine aggregates three analysis engines: (1) **GeneSentenceAnnotator**, (2) **GeneNameAnnotator**, and (3) **GeneTagAnnotator**. It uses a fixed flow to pass the annotations in the CAS from its constituent analysis engines in the order specified above. As a whole, it encompasses the functionality of taking in input by the SentenceCollectionReader, breaking up the sentences into their id and word elements, and then annotating them using the GeneNameAnnotator and GeneTagAnnotator. Finally, it outputs annotated GeneSentences, Words, and GeneTags that may be used by the CAS consumer to process the correct output.

3. GeneAnnotationPrinter:

   Based on the UIMA examples AnnotationPrinter, this simple CAS consumer merges the GeneSentence, Word, and GeneTag annotation to process the correct output. It uses the Words to match the non-whitespace offsets from the beginning of the sentence with the corresponding gene terms. It then uses the id stored in the GeneSentence annotations to pair the sentences with the corresponding GeneTerms. Finally, it writes the correct output to file (`hw1-amaiberg.out`).

1.1 GeneSentenceAnnotator

GeneSentenceAnnotator is a simple analysis engine that takes a sentence from collection reader and outputs the SentenceId, constituent Words, and non-whitespace character offsets from the beginning of the sentence. As such, this analysis engine can essentially be seen as a pre-processing tokenization step.

1.2 GeneNameAnnotator

GeneNameAnnotator takes a sentence containing gene mentions and runs the PosTagNamedEntityRecognizer to find relevant gene terms. Unfortunately, since performance was better with the GeneTagAnnotator alone, this analysis engine is currently superfluous. However, it may prove to be more useful if other components were added to the pipeline in the future. It is still included as part of the pipeline to demonstrate functionality.

1.3 GeneTagAnnotator

GeneTagAnnotator takes sentences as input, and uses LingPipe to run the genetag.HmmChunker (a Hidden Markov-Model chunker) on them to extract gene-related chunks. It then ouputs these chunks into the CAS. This chunker is pretrained on the GENETAG corpus, and is provided on the LingPipe tutorial pages[1].

2 Type System

The type system as described by the (unoriginally titled) typesystemDescriptor.xml, is shared across all the components in the CollectionProcessingEngine, and defines the basic types used for each analysis engine. The basic types are:

- GeneSentence:

  GeneSentence contains references (or features) to the `gene_id` or the identification string corresponding to the sentence, and a `sentence` containing the actual content of the sentence. These types are used by the GeneSentenceAnnotator to separate the id from the actual content of the sentence for further processing in the pipeline.

- Word

  Word contains features `beginOffset` corresponding to the non-whitespace offset of the word from the beginning of the sentence, and `content` corresponding the actual content of the word. This type is used by GeneSentenceAnnotator to tokenize the sentences and store their non-whitespace offsets for further processing in the pipeline.

- GeneTerm:

  GeneTerm contains the feature `term` corresponding to the term identified by the Stanford-CoreNLP pipeline. This type is used by GeneNameAnnotator to identify relevant gene terms and store them in the CAS.

- GeneTag:

  GeneTag contains the feature `tag` corresponding to the gene mention chunk identified by the LingPipe genetag.Hmm.chunker. GeneTagAnnotator uses this type to store relevant gene chunks and store them in the CAS.

[1] see: `http://alias-i.com/lingpipe/demos/tutorial/ne/read-me.html`