ISMLA 17/18: UIMA Exercises

1 Aggregate Analysis Engine

In this exercise you will write a second primitive analysis enginge to combine with the sentence detector to an aggregate analysis engine.

1.1 Tokenization AE

Set up a wrapper around the OpenNLP tokenizer:

1. Create a simple AE descriptor and link it to a corresponding Java AE class.
2. Set `Sentence` as input and `Token` as output capability.
3. Iterate over `Sentence` annotations:
   - for each sentence, get the token spans
   - annotate `Token` types; use `tokenizePos` as a basis for obtaining spans

1.2 Aggregate Analysis Engine Descriptor

1. Create an aggregate analysis engine descriptor.
2. Add the sentence and token detector, ensure they are executed in the correct order.
3. Test your aggregate AE with the DocumentAnalyzer on last session's data.

2 CollectionReader

1. Set up a collection reader descriptor file with two String-valued configuration parameters: language and inputFile.
2. Write a collection reader that gives back a JCas for each line of an input file described by the configuration parameter.