11-791 Design and Engineering of Intelligent Information System Fall 2012

UML Design and Named Entity Recognition Implementation with UIMA SDK
HOMEWORK 1

NISARGA MARKANDAIAH
nmarkand@cs.cmu.edu

PART 1: SOFTWARE DESIGN

1. UML DESIGN: The UML employed in the development of the system is shown below:

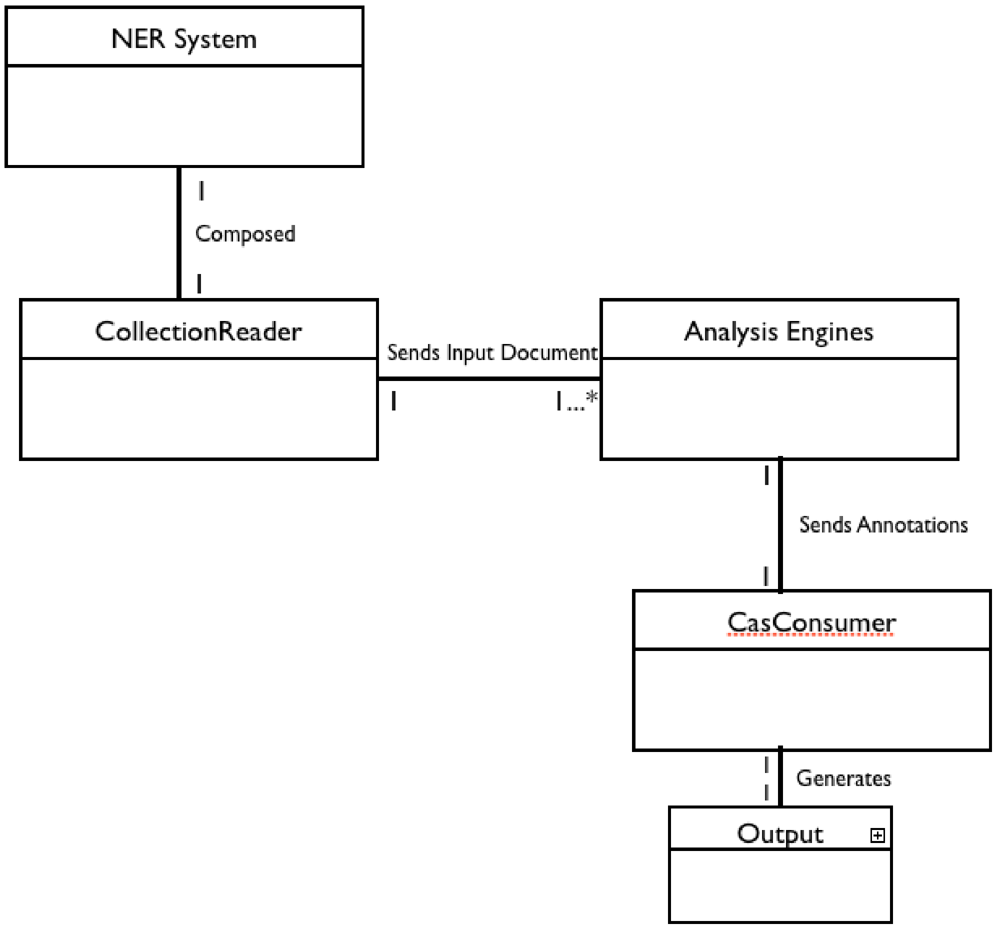

Figure 1: Domain Model for NER system

PART 2: SYSTEM DESIGN

1. ARCHITECTURAL STYLES: The system was built using a form of **incremental** method. Where each component was first built and tested. This model combines the elements of the waterfall method along with iterative philosophy of prototyping.

2. GENERAL FLOW OF SYSTEM

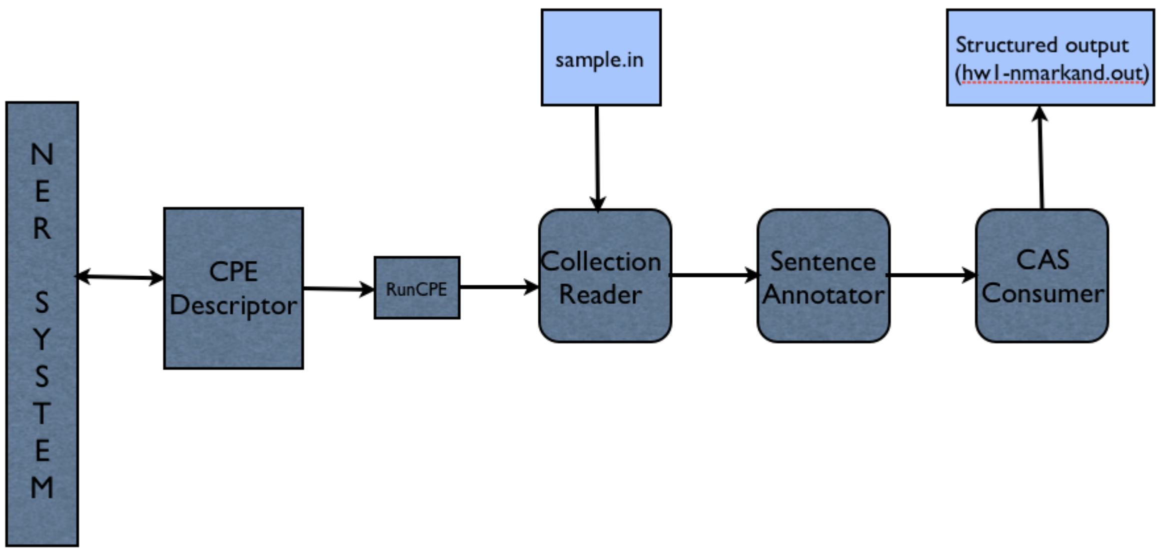

Figure 2: General pipeline followed by the system

<u>3. SUBSYTEMS:</u> The subsystems in the system are as follows:

1. CollectionReader is represented below:

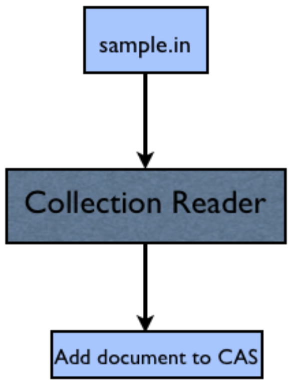

Figure 3: Collection Reader pipeline

2. SentenceAnnotator
The sentence annotator was the most important module in the system where LingPipe has been integrated. Each of the sentence is taken for the input document, and given as an input to the chunker HMM.

There is a list of gene locations returned, which are stored in a list, parsed accordingly and gene names retrieved. There was also the need to have a special module to remove the spaces, for the beginning offset which we calculated.
The type system was created and the retrieved annotations were appended as features to this system.
This process is elaborated in the diagram given below:

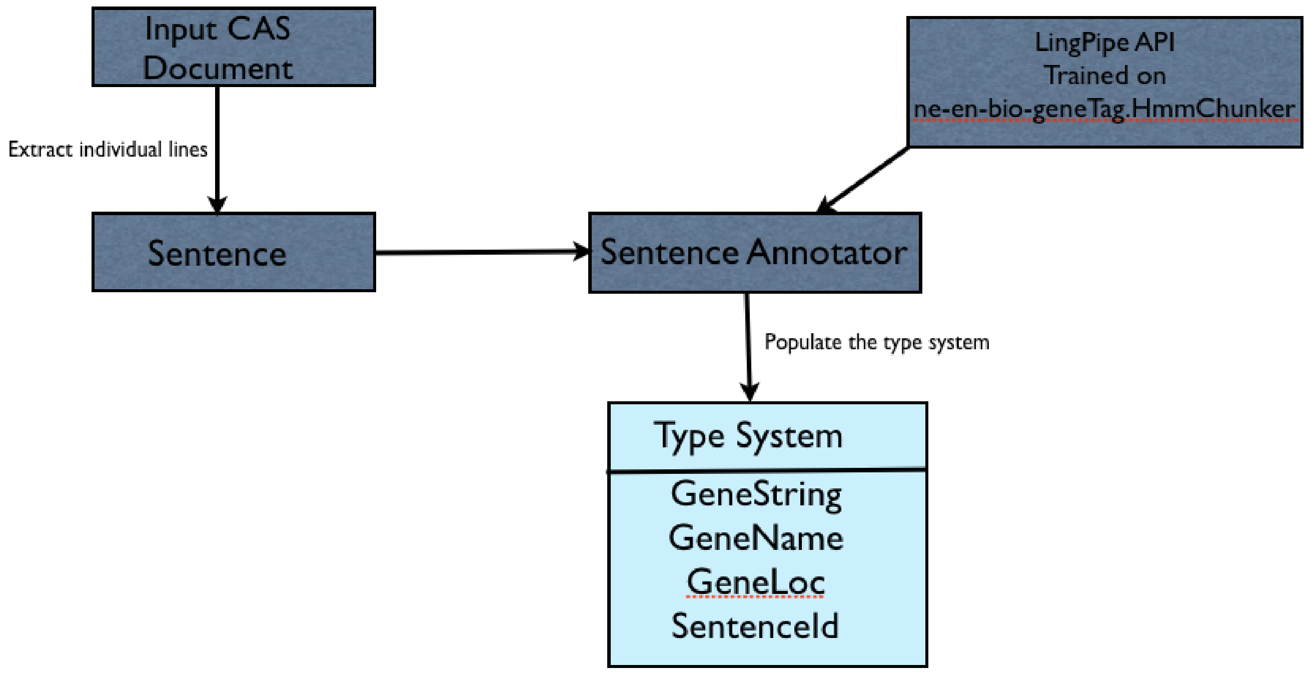

Figure 4: Sentence Annotator pipeline

3. CasConsumer

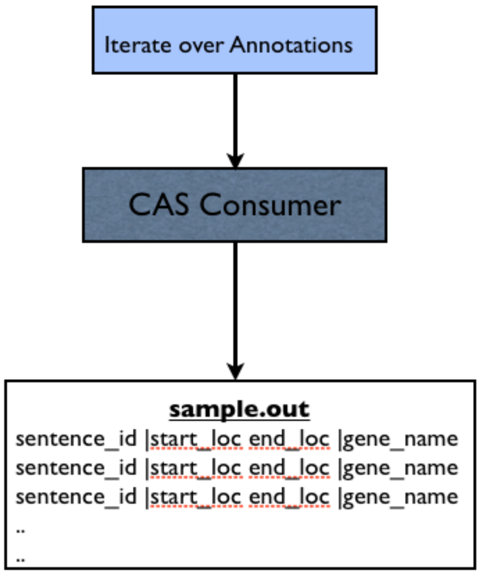
Iterate over Annotations
CAS Consumer
sample.out
sentence_id |start_loc end_loc |gene_name
sentence_id |start_loc end_loc |gene_name
sentence_id |start_loc end_loc |gene_name
..
..

Figure 5: Cas Consumer Pipeline

PART 3: DETAILS OF IMPLEMENTATION

1. EXTERNAL LIBRARIES

A variety of approaches was examined for building an efficient NER system. These were rule based approaches, HMM's, CRF's. Also 2 external libraries were examined, which are stable systems in the biomedical analysis field- Abner and LingPipe. LingPipe was chosen amongst the 2, since it showed a better overall performance.

2. ALGORITHMS USED

The LingPipe NER was built using a statistical hidden markov model, where the words in the corpus are divided into non recursive chunks of named entities, and correspondingly given weights. The given input file is run on the already trained system. A HMM based NER chunker is known to have better performance than handcrafted rules.
(Source: http://acl.ldc.upenn.edu/acl2002/MAIN/pdfs/Main036.pdf)

3. EXTERNAL TRAINING DATA

The training data used in the training of the LingPipe NER system, is provided by the Unites States National Center for Biotechnology Information and is called as GENETAG: a tagged corpus for gene/protein named entity recognition. A Hmm chunker model is built out of this corpus and used in the system.

4. DATA STRUCTURES / TYPE SYSTEM

The type system called sentence annotation was created which has various features. These are briefly explained below:
1.GeneName: The name of the gene found in a given string
2.GeneString: The corresponding sentence containing the gene
3.SentenceId: The ID of the sentence string containing the particular gene
4.GeneLoc: The starting location of the gene under consideration.

(the end location of the gene wasn't added as a feature, since it can easily be deducted considering the length of the gene and the starting location.)

PART 4: RESULTS & ANALYSIS

RESULTS:

A results were analysed in terms of precision and recall.

- Total number of retrieved annotations= 20174
- Total number of correct annotations retrieved=14442
- Total gold sample annotations=18265

- Precision: 14442/20174=**0.72**
- Recall: 14442/18265= **0.79**

ANALYSIS:

We notice that the system integrated with the LingPipe API gives us a decent score for precision and recall. There are a variety of ways we can improvise the system:
1. Use another commonly available API, calculated confidences in intermediate steps, and finally obtain the genes with the highest confidence scores.
2. Also, we can use simple rule based ideas, to get an overall improvement in the system, especially when there are gene names which are abbreviated.