

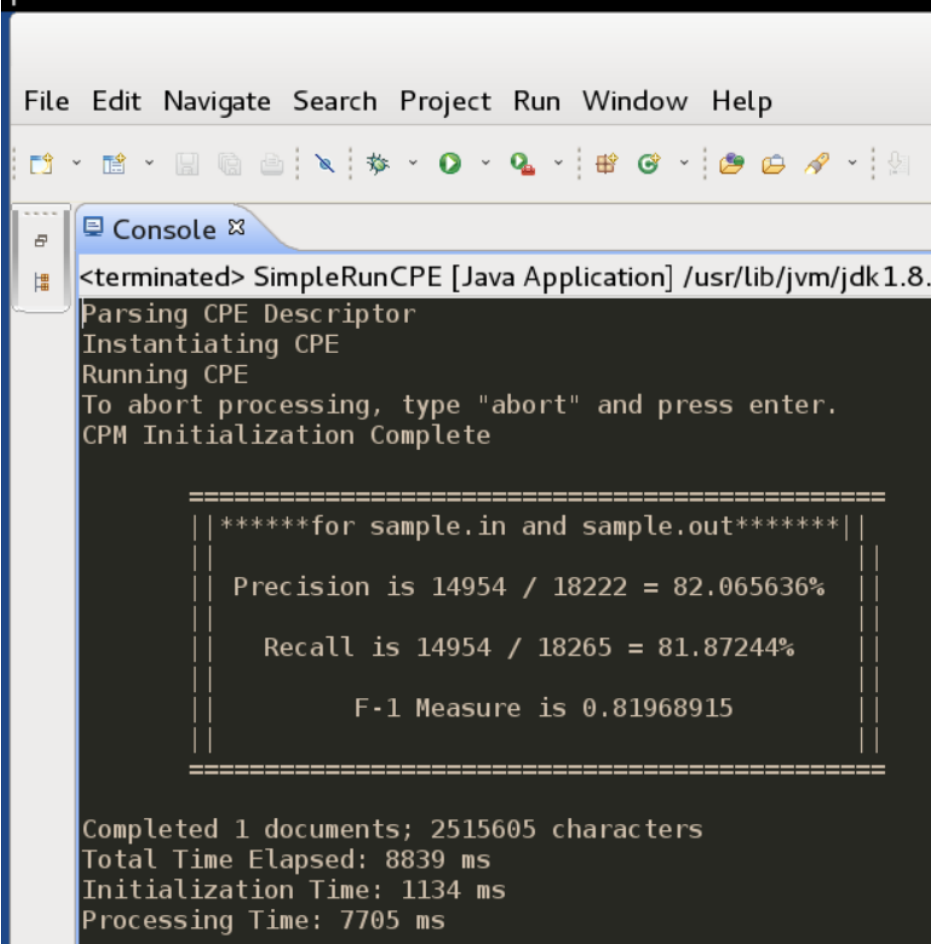
Name: Long He

Andrew ID: longh

**1. Please identify/describe any machine learning techniques used:.....**

I construct a statistical named entity recognizer by importing a model file called ne-en-bio-genetag.HmmChunker and reconstituting a ConfidenceChunker, which provides chunking based on a hidden Markov model (HMM) language models. It runs chunking in such a way as to return chunks in order of confidence, which I take to be the probability of the chunk given the input text,  $P(\text{chunk}|\text{text})$ . Then I walk over the arguments, extracting a character array as before, and then providing it to the chunker, this time calling the nBestChunks method. With a confidence chunker, the result iterator is over chunks, so the cast is to (Chunk) in the body of the iteration. In this case, I use grid search manually to select the maximum n-best chunks to a constant (30) and the confidence parameter for the language models (0.6).

The preliminary result is precision 14954/18222 (82.1%), recall 14954/18265 (81.9%) and F-1 measure (0.820) and the output file hw1-longh.out is at the project root.



```
<terminated> SimpleRunCPE [Java Application] /usr/lib/jvm/jdk1.8.  
Parsing CPE Descriptor  
Instantiating CPE  
Running CPE  
To abort processing, type "abort" and press enter.  
CPM Initialization Complete  
  
*****for sample.in and sample.out*****  
Precision is 14954 / 18222 = 82.065636%  
Recall is 14954 / 18265 = 81.87244%  
F-1 Measure is 0.81968915  
  
Completed 1 documents; 2515605 characters  
Total Time Elapsed: 8839 ms  
Initialization Time: 1134 ms  
Processing Time: 7705 ms
```

I also used Stanford POS to do this and find its performance bad. It has about 10% precision and 50% recall and have little room to improve regardless of my efforts to exclude more wired words.

**2. Please identify/describe any NLP techniques/components used:.....**

Some NLP techniques I used are covered in the previous answer. In addition, I exclude all the single character words to improve the performance.

**3. Please identify/describe any external (marked up text) training data used:.....**

I import a model file called ne-en-bio-genetag.HmmChunker from <http://alias-i.com/lingpipe/demos/tutorial/ne/read-me.html>.

**4. Please identify/describe any external lexical resources (terminology lists)used:.....**

No.

**5. Please describe any rule sets used:.....**

No.

**6. If your system interacts with or uses data from any biological database(s), please describe:.....**

No.

**7. Please identify/describe any other relevant resources used to train/develop your system:.....**

No.

**8. Please describe the general data flow in your system:.....**

Collection Reader reads the file as a whole string.

Then, SentenceAnnotator splits the string into individual sentences and adds ids and texts of sentences as annotations.

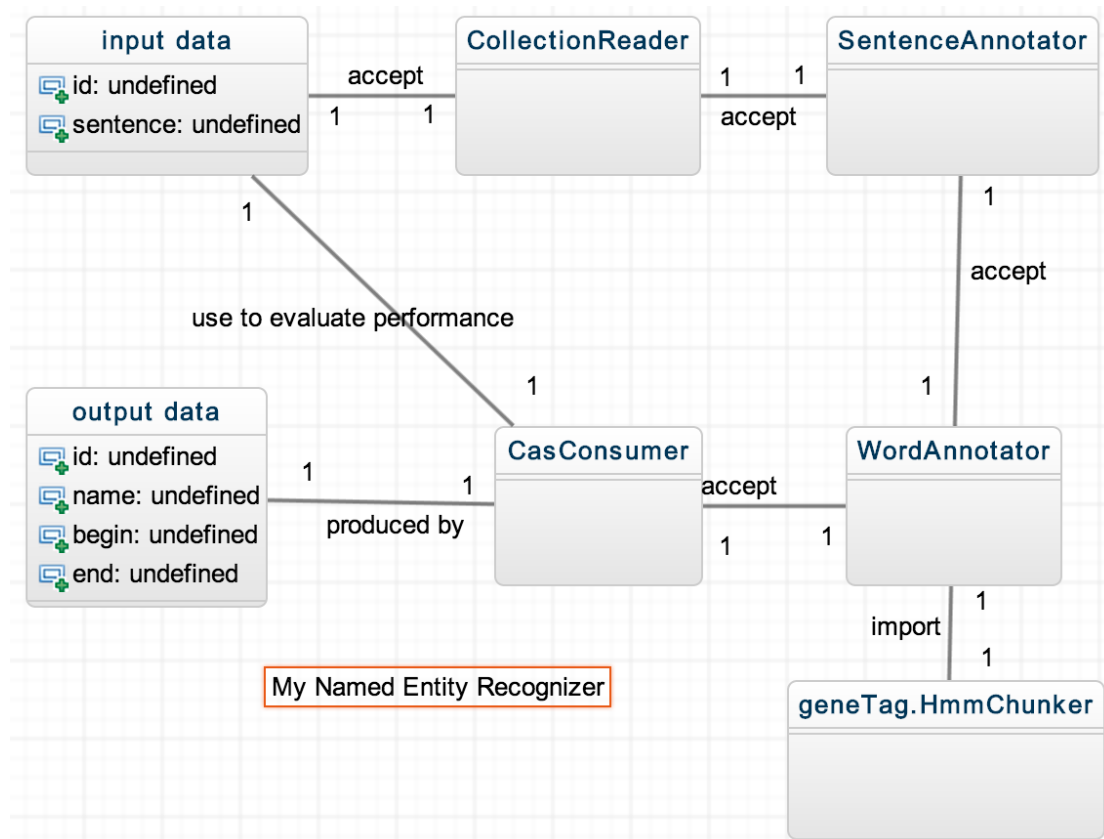
Next, WordAnnotator splits the sentences as words and uses a HMM model to locate these gene words. It also adds id, begin (offset counted), end (offset counted), name of gene words as annotations.

Finally, CasConsumer writes all these annotations into the output file and calculates the precision, recall and F-1 measure.

## 9. Other information of interest:.....

I find that if I exclude all the single character words, the precision, recall and F-1 measure become higher.

This picture below is the UML of my NER:



References:

<http://alias-i.com/lingpipe/demos/tutorial/ne/read-me.html>