Homework 2
11-791

Name: **Shriphani Palakodety**
Andrew ID: **spalakod**

October 10, 2014

1 Task Description

In this task the goal was to implement and improve the performance of a gene NER tagging task. The model described in this paper combined a statistical approach (an HMM model) and a dictionary approach. The intuition is that combining two distinct approaches with two different datasets makes the model robust to overfitting. The model was chosen after a set of experiments incorporating other approaches. The experiments, datasets and results are described in the following sections.

2 System Description

The type system used extends the provided types with two additional types: (i) `Sentence`, and (ii) `NERAnnotation`. This type hierarchy is shown in Figure 1.

The submitted pipeline contains of three annotators. The flow is shown in Figure 2.

The `SentenceAnnotator` takes the supplied lines from the file and adds `Sentence` annotations (a mapping from IDs to text in the provided corpus) to the `CAS`. The `LingpipeNERAnnotator` employs a HMM trained on the Genetag corpus [2] and adds `NERAnnotation`s (begin and end-points of annotations) to the `CAS`. The `DictionaryAnnotator` leverages a curated dictionary (details provided later) to add more `NERAnnotation`s. Finally, a union of the annotations produced by these two systems is written as output. The final UML diagram in shown in Figure 3.

3 Experiments and Evaluated Pipelines

In this section the approach used to converge to the final pipeline is described. At first, the idea was to combine two distinct approaches for producing the annotations: (i) statistical (leveraging a HMM model or a CRF model), and (ii)

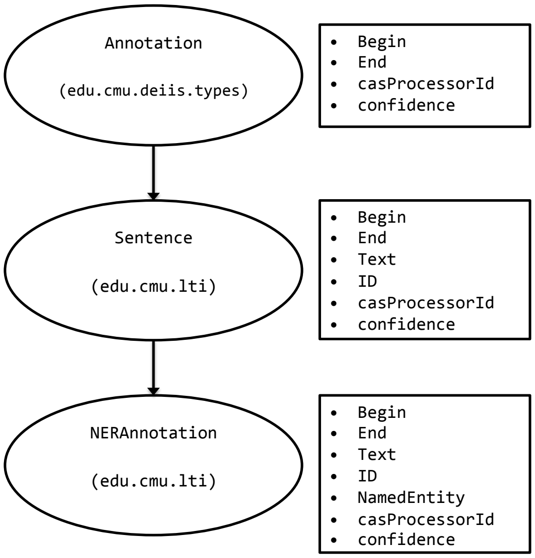

Figure 1: The type system used in the pipeline

Precision	0.768513928819
Recall	0.84883657268
F1	0.806680714899

Table 1: Results for the Lingpipe NER tagger pipeline

dictionary based. This allows our model to not overfit one particular dataset and also allows us to incorporate two approaches.

3.1 Statistical Approach

The two approaches considered were (i) HMM (from the Lingpipe library [1]), and (ii) CRF (from the Abner library). The HMM model was trained on Genetag and the CRF model was trained on a subset of the GENIA corpus (NLPBA specifically) and another CRF model was trained on the BioNER corpus. The systems were then evaluated on the data provided in HW1. In addition, some experiments were performed in incorporating additional hypotheses from the HMM model.

Table 1 shows the results for the simple HMM model from Lingpipe trained on the Genetag corpus:

Table 2 shows the results for the CRF model from ABNER trained on the

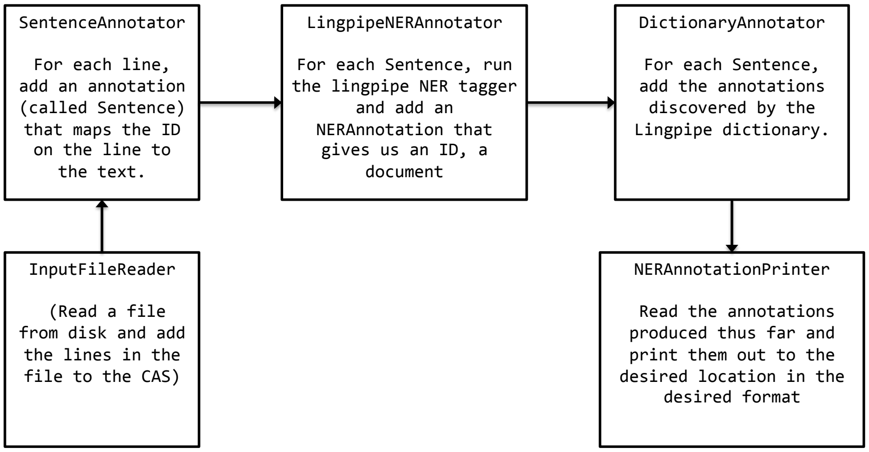
SentenceAnnotator
For each line, add an annotation (called Sentence) that maps the ID on the line to the text.
LingpipeNERAnnotator
For each Sentence, run the lingpipe NER tagger and add an NERAnnotation that gives us an ID, a document
DictionaryAnnotator
For each Sentence, add the annotations discovered by the Lingpipe dictionary.
InputFileReader
(Read a file from disk and add the lines in the file to the CAS)
NERAnnotationPrinter
Read the annotations produced thus far and print them out to the desired location in the desired format

Figure 2: The pipeline itself

Precision	0.41492358129
Recall	0.491979195182
F1	0.450177846801

Table 2: Results for ABNER on NLPBA

GENIA (NLPBA) corpus and Table 3 shows the results when the model is trained on the NLPBA corpus.

Clearly, the CRF model performs poorly compared to the HMM model. This can be attributed to (i) poor quality of training annotations (as evidenced by the poor precision) in the GENIA corpus, (ii) insufficient diversity of annotated content in the corpus (as evidenced by the inferior recall). As a result the HMM model was chosen for the statistical module.

After choosing the model, some evaluation was performed on incoporating alternative hypotheses rather than the top produced hypotheses. The results are shown in Tables 7, 6, 5, 4. The conclusion drawn was that adding additional hypothesis significantly reduces the precision without producing any significant gains in the recall. Thus only the top result was obtained and used.

Though the final pipeline doesn't include the ABNER and alternate HMM models, they are still provided in the submitted jar file. Please see the Javadoc and the source to see these implementations.

Precision	0.173944941308
Recall	0.545195729537
F1	0.263742666826

Table 3: Results for ABNER on BioNER

Precision	0.246475504404
Recall	0.959102107857
F1	0.392169153449

Table 4: Adding 5 Alternate Hypotheses

Precision	0.292374881964
Recall	0.949301943608
F1	0.44706003687

Table 5: Adding 4 Alternate Hypotheses

Precision	0.292374881964
Recall	0.949301943608
F1	0.44706003687

Table 6: Adding 3 Alternate Hypotheses

Precision	0.494198845719
Recall	0.909499041883
F1	0.640413269338

Table 7: Adding 2 Alternate Hypotheses

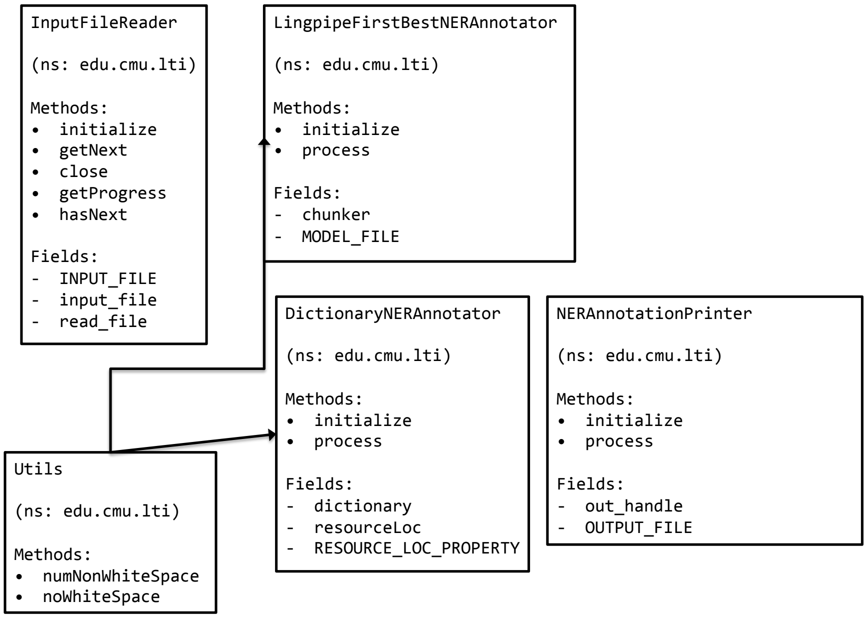

Figure 3: The final UML diagram

Precision	0.759640291286
Recall	0.850971803997
F1	0.8027165212

Table 8: Results of the final pipeline

3.2 Dictionary Approach

The HGNC [3] website contains a list of Gene names. From this, a list of gene symbols and names was obtained and a dictionary was built. The script used for this task is available in `src/main/resources/build_dictionary.py`. Two dictionaries were evaluated (i) including actual gene symbols, and (ii) not including gene symbols. It was onserved that the annotations provided in the data from HW1 did not contain symbols but just names. Thus we only used the dictionary not containing the symbols.

4 Final Pipeline Results

The final pipeline performance is shown in Table 8.

5 Conclusions

In this document, the experiments conducted to produce an NER tagger for the gene-mention detection task are presented. The final pipeline consited of an HMM model and a dictionary based approach for obtaining NER mentions. All results reported were on the dataset provided in HW1.

References

[1] Alias-i. LingPipe 4.1.0. http://alias-i.com/lingpipe (accessed September 21, 2014)

[2] Tanabe, L., Xie, N., Thom, L. H., Matten, W., Wilbur, W. J., GENETAG: a tagged corpus for gene/protein named entity recognition BMC Bioinformatics

[3] Gray KA, Daugherty LC, Gordon SM, Seal RL, Wright MW, Bruford EA. genenames.org: the HGNC resources in 2013. Nucleic Acids Res. 2013 Jan;41(Database issue):D545-52.