11791 Design and Engineering of Intelligent Information Systems

Assignment 1 Report

Name: LIU Xi

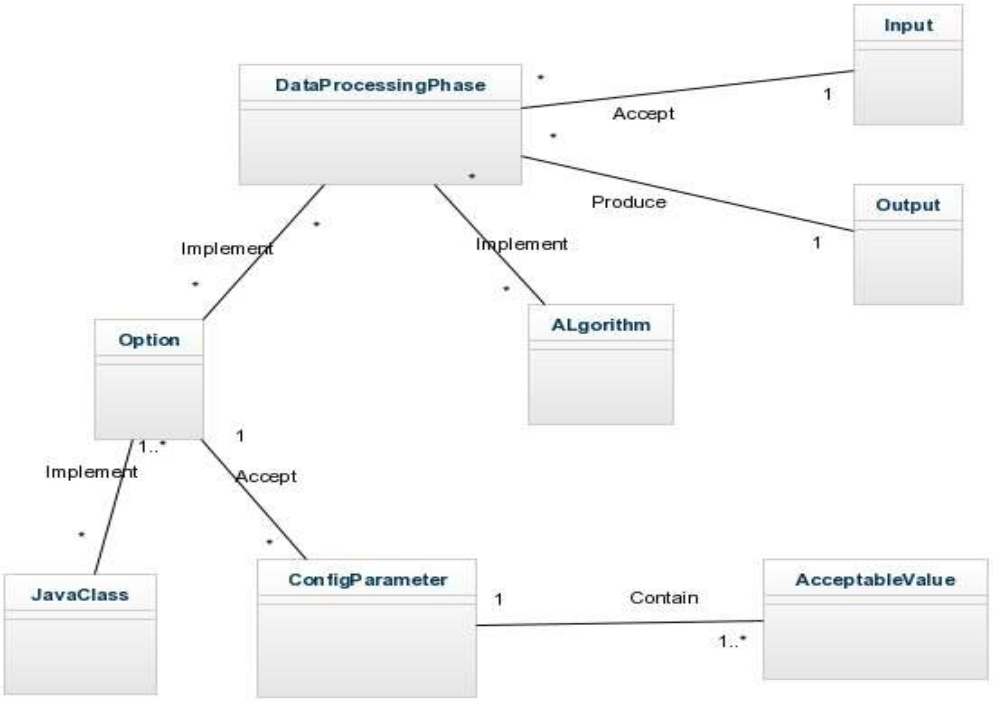
Andrew ID: xiliu1

Date: 09-23-2014

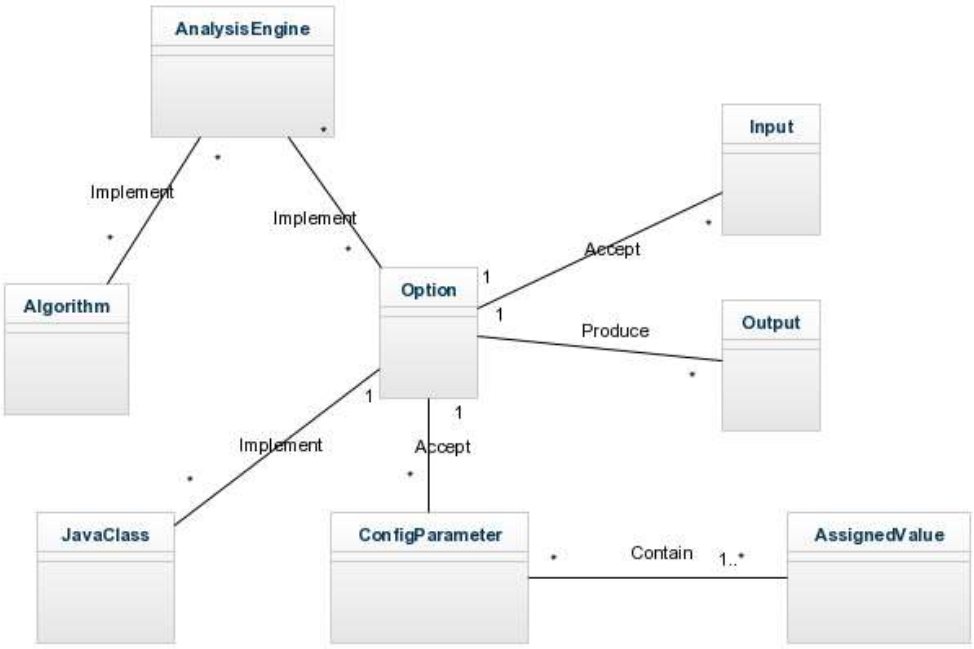
Contents

1. Domain Model Diagram for Task 1

IntellignentInformationSystem



AnalysisEngine



2. External Resources

Machine Learning Techniques

Hidden Markov Model
HMM is a simplest dynamic Bayesian network, where the state is not hidden while the output relying on the state is visible. For each possible output tokens, the states have probability distributions, so that the sequence of tokens shows the sequence of states in HMM. HMM is widely used for natural language recognition, and also solve the bio-informatics tagging problems in a linguistic approach.

Tools

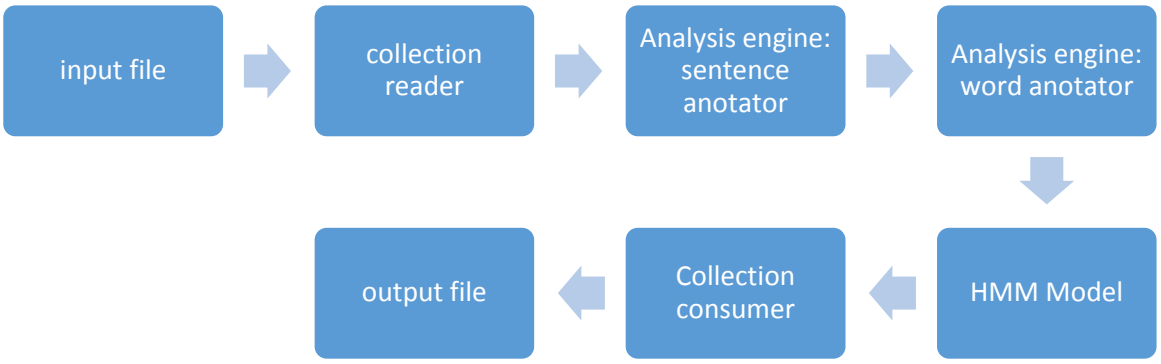
Toolkit: LingPipe
LingPipe is toolkit for processing text using computational linguistics with Java.
LingPipe Source Code: GeneTag Corpus

Dataset: NCBI GeneTag
Produced by the Unites States National Center for Biotechnology Information (NCBI),
Related paper: Tanabe, L., N. Xie L. H. Thom, W. Matten, And W. J. Wilbur. 2004. GENETAG: a tagged corpus for gene/protein named entity recognition. BMC Bioinformatics 2005, 6(Suppl 1):S3.
Download path: ftp://ftp.ncbi.nlm.nih.gov/pub/lsmith/MedTag/medtag.tar.gz

3. General Data Flow

Scenarios:
Step 1. Load the input and break the input file by sentences
Step 2. Break the sentences by words
Step 3. Predict if the word is a gene tag
Step 4. Calculate the position of gene tag and output

4. NER Pipeline Design



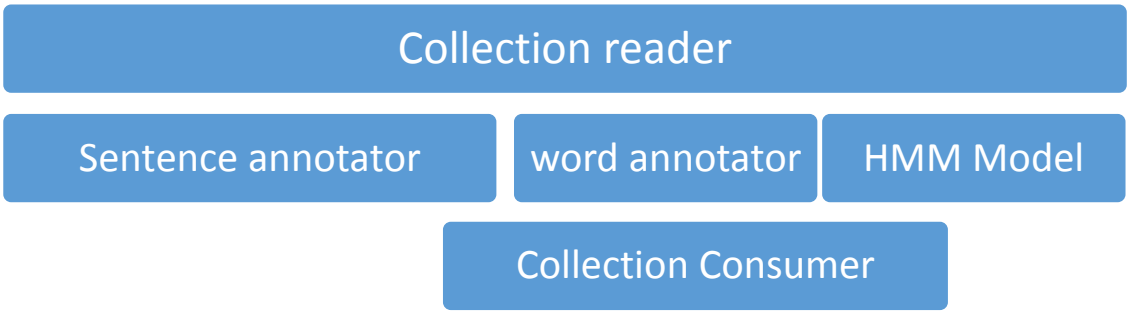
5. NER System Design

Type System

Sentence: represents the sentences with two parameters, source and confidence.
Word: represents the segmented words with two parameters, source and confidence.

Collection Processing Engine

Collection Processing Engine implements the pipeline design. The function of each parts are described later.



Collection Reader

Scenarios:
Step 1. Load the input file.
Step 2. Break the input file by sentences, and record the absolute position of each sentence, and feed the sentence annotator.

Analysis Engines

There are two annotators in total, and both read and produce JCAS data.

Sentence Annotator

Scenarios:
Step 1. Read the sentences, and break the sentence into words, and record the absolute position of each word,
Step 2. Feed the word to HMM model.

Word Annotator

Scenarios:
Step 1. Read the word
Step 2. Apply HMM model to find the gene tag.
Step 3. Feed the predicted gene tag to collection consumer.

CAS Consumer

The CAS Consumer read JCAS data from annotators and output the results to the output file.

6. Reflections

Precision

The sample built by Stanford-NLP only recognize the nouns, so the precision, around 10% is very low.
The proposed LingPipe solution implements the HMM trained on the gene tags datasets, which is more capable for gene tag recognition and achieved the precision at around 80%.

Problems

The major difficulty in this project is setting up the development environment and interact with UIMA Toolkit GUI. Sometimes the unstable network connection also interrupt building the archetype.

Further Enhancements

The project could be further enhanced by trying more machine learning models, and improve the performance of current HMM.