CheNER

Chemical Named Entity Recognizer User Guide

(2012)

CheNER is a named entity recognition tool, that uses Conditional Random Fields for identifying mentions of chemicals in text, focusing on IUPAC entities.

<u>Usage:</u>

To run the user interface:

- from the command-line execute: ***java -jar CheNER.jar***
- double-click on the jar file (in Windows operating system)

To run an example of how to tag a simple string text, please execute from the command-line:

java -cp paht/to/CheNER.jar Examples.TaggingExample

To run an example of how to train a customazed dataset, please execute from the command-line:

1. ***java -cp path/to/CheNER.jar Examples.TrainingExample 'trainFile' 'modelFile'***
2. ***java -cp path/to/CheNER.jar Examples.TrainingExample 'trainFile' 'modelFile' 'p1,p2,p3,p4,p5,p6,p7,p8'***

   where p1 indicates if MF feature class is activated (1) or not (0), is applied the same logic for the rest of pN in order for ToC, PS, W, WB, L, BNS and SW

To use CheNER in your code:

```
String inputFile="Glutamic acid analogs containing 3- and 4-methyl and 2-, 3-,"
        + "and 4-phenyl substituents were prepared.";
Tagger t = new Tagger();
//type=0 -> identify IUPAC type entities
//type=1 -> identify All type entities
t.doTheTagging(inputFile, "annotatedText.txt", 0, false);
Hashtable entities=t.getEntities();
Enumeration en=entities.keys();
  while(en.hasMoreElements()){
    String entity=en.nextElement().toString();
    int n=Integer.parseInt(entities.get(entity).toString());
    System.out.println(entity+" ("+n+")");
    }
  }
```

To use CheNER from shell/commandLine:

```
> java -cp CheNER.jar chener.CommandLineClass
```

Class Summary:

Tagger	This class uses the CRF interface to do the named entity tagging.
Trainer	This class will train the CRF to extract the entities from the customized dataset.
Input2Tokens	This class tokenize the text into words, and labels only in the training process.
Feature4Tokens	This class extract the different types of features defined for each word.

Tagger Class Detail:

This class uses the CRF interface to do the named entity tagging.

By default all methods use a tokenization by spaces. The tokenization can't be modified.

Constructor Detail
public Tagger() Basic constructor. Will use the CheNER's default features set (MF, ToC, PS, W, WB, L, BNS, SW)
public Tagger(String modelFile, List<String> parameters) External constructor. Load a trained CRF specified by the external *modelFile*. *Parameters* list define which feature classes from the default feature set were activated to use in the creation of the *modelFile*.
Method Detail
public void doTheTagging(String inputFile, String outputFile, int type, boolean isFile) Takes input text that can be in two formats, if *isFile* is true the *inputFile* is a file name, else the *inputFile* is just an string text. The text is annotated and saved in the corresponding output *outputFile*. The model used to tag the text is defined by *type*. If *type* is equal to 0, is used the IUPAC type entities. If *type* is equal to 1, is used the All type entities. Else is used the customized train model.
public void doTheTagging(String[] inputFile, String outputFile, int type) Takes input text from *inputFile* that contains a list of file names, and is annotated and saved in the *outputfile*.The model used to tag the text is defined by *type*. If *type* is equal to 0, is used the IUPAC type entities. If *type* is equal to 1, is used the All type entities. Else is used the customized train model.
public Hashtable getEntities(); Returns the entities found (hashtable keys) in the input text and their ocurrences (hashtable values).

Trainer Class Detail:

This class will train the CRF to extract the entities from the customized dataset. The input file or training file must be tokenized with one word per line, with a \tab separating the word/token from its label.The first token of an entity name should have a label "B", all other entity token labels should be "I", and non-entity tokens should be labeled with "O", following the IOB scheme. If the input file contains more than one document, each document must be separatet by a new white line.

```
The	O
synthesis	O
of	O
a	O
range	O
of	O
3-hydroxy-4(1H)-pyridinones	B
```

Constructor Detail
public Trainer()
Method Detail
public void doTheTraining(String trainFile, String modelFile) Takes input *trainFile* which format is described above, and saves the trained linear-chain CRF of second order using CheNER's default feature set in the corresponding output *modelFile*.
public void doTheTagging(String trainingFile, String modelFile, List<String> parameters) Takes input *trainFile* which format is described aboive, and saves the trained linear-chan CRf of second order in the corresponding output *modelFile*, using the *parameters* list that defines which feature class of the CheNER's default feature set are activated or not to use.