How to train a model for Sentence Detection in openNLP

How to train a model for Sentence Detection in openNLP

How to train a model for Sentence Detection in openNLP– In this tutorial, we shall understand how to train a model from input training data for Sentence Detection in openNLP using Java.

Why to train a model for Sentence Detection

There would always be a requirement for sentence detection. There could be new structure of statements in your use case or may be sentence detection has to be done for a language different from English or something that is readily not available. These scenarios would call out to build a model of our own, from our own training data, for our own purpose.

Train a model for Sentence Detection

Now let us see how to train a model for Sentence Detection in openNLP. Follow the below steps:

1. Create a text file and keep a sentence for each line in the text file.
2. Create an InputStreamFactory from the input file using code snippet shown below.
3. Set the machine learning hyper parameters like number of iterations and cutoff using the code snippet shown below.
4. With the help of train() method in SentenceDetectorME, generate a model.

Let us generate a model file with the help of training, as shown in below example:

Sentence Detector Training Example in openNLP

The following example SentenceDetectorTrainingExample.java shows how to train a model for your own training data. If you would like to know how to setup java project to use openNLP, in eclipse, refer to setup of java project with openNLP libraries, in eclipse. The process should be same, even for a different IDE(adding the required jars to the build path should do the magic).

Download ? SentenceDetectorTrainingExample.java & trainingDataSentences.txt

SentenceDetectorTrainingExample.java

```
import java.io.File;
import
```

```
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.nio.charset.StandardCharsets;

import opennlp.tools.sentdetect.SentenceDetectorME;
```

```
import opennlp.tools.sentdetect.SentenceModel;
import opennlp.tools.sentdetect.SentenceSampleStream;
import opennlp.tools.util.InputStreamFactory;
import opennlp.tools.util.MarkableFileInputStreamFactory;
import opennlp.tools.util.PlainTextByLineStream;
import opennlp.tools.util.TrainingParameters;

/**
* @author tutorialkart
*/
public class SentenceDetectorTrainingExample {

   public static void main(String[] args) {
      try {
         new SentenceDetectorTrainingExample().trainSentDectectModel();
      } catch (IOException e) {
         e.printStackTrace();
      }
   }

   /**
    * This method generates s custom model file for sentence detection, in directory "custom_models".
    * The training data used is "trainingDataSentences.txt". Training data contains a sentence per line in the text file.
    * @throws IOException
    */
   public void trainSentDectectModel() throws IOException {
      // directory to save the model file that is to be generated, create this directory in prior
      File destDir = new File("custom_models");

      // training data
      InputStreamFactory in = new MarkableFileInputStreamFactory(new File("trainingDataSentences.txt"));

      // parameters used by machine learning algorithm, Maxent, to train its weights
      TrainingParameters mlParams = new TrainingParameters();
      mlParams.put(TrainingParameters.ITERATIONS_PARAM, Integer.toString(15));
      mlParams.put(TrainingParameters.CUTOFF_PARAM, Integer.toString(1));

      // train the model
      SentenceModel sentdetectModel = SentenceDetectorME.train(
            "en",
            new SentenceSampleStream(new PlainTextByLineStream(in, StandardCharsets.UTF_8)),
            true,
            null,
            mlParams);

      // save the model, to a file, "en-sent-custom.bin", in the destDir : "custom_models"
      File outFile = new File(destDir,"en-sent-custom.bin");
      FileOutputStream outFileStream = new FileOutputStream(outFile);
```

```
        sentdetectModel.serialize(outFileStream);

        // loading the model
        SentenceDetectorME sentDetector = new SentenceDetectorME(sentdetectModel);

        // detecting sentences in the test string
        String testString = ("Sugar is sweet. That doesn't mean its good.");
        System.out.println("\nTest String: "+testString);
        String[] sents = sentDetector.sentDetect(testString);
        System.out.println("---------Sentences Detected by the SentenceDetector ME class using the generated model-----
--");
        for(int i=0;i<sents.length;i++){
            System.out.println("Sentence "+(i+1)+" : "+sents[i]);
        }
    }
}
```

Download ? SentenceDetectorTrainingExample.java & trainingDataSentences.txt

When SentenceDetectorTrainingExample.java is run, the output to console is :

Program Output

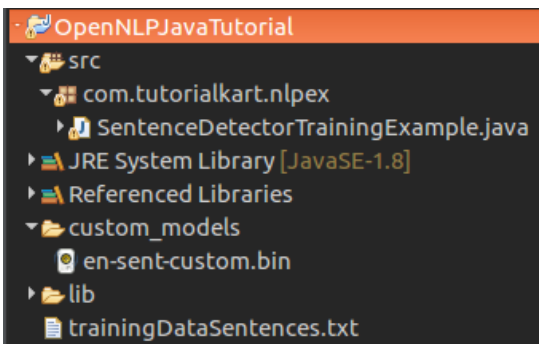
```
Indexing events using
cutoff of 1
```

```
Indexing events using cutoff of 1

    Computing event counts... done. 128 events
    Indexing... done.
Sorting and merging events... done. Reduced 128 events to 128.
Done indexing.
Incorporating indexed data for training...
done.
    Number of Event Tokens: 128
        Number of Outcomes: 2
      Number of Predicates: 279
...done.
Computing model parameters ...
Performing 15 iterations.
  1: ... loglikelihood=-88.72283911167311 0.890625
  2: ... loglikelihood=-40.857996455731566 0.90625
  3: ... loglikelihood=-32.22640634368208 0.90625
  4: ... loglikelihood=-27.13613120396953 0.90625
  5: ... loglikelihood=-23.386731336945246 0.90625
  6: ... loglikelihood=-20.51509016196713 0.9140625
  7: ... loglikelihood=-18.262162454424875 0.9296875
  8: ... loglikelihood=-16.453775397116225 0.9453125
  9: ... loglikelihood=-14.972158154339848 0.9609375
 10: ... loglikelihood=-13.736687458210751 0.9921875
 11: ... loglikelihood=-12.690904850490426 1.0
 12: ... loglikelihood=-11.794274308551026 1.0
 13: ... loglikelihood=-11.016996711268375 1.0
 14: ... loglikelihood=-10.336696500866838 1.0
 15: ... loglikelihood=-9.736254071469505 1.0

Test String: Sugar is sweet. That doesn't mean its good.
---------Sentences Detected by the SentenceDetector ME class using the generated model-------
Sentence 1 : Sugar is sweet.
Sentence 2 : That doesn't mean its good.
```

The project structure, training input file location and model file generation location, etc., for the example is shown below:



Conclusion :

In this openNLP tutorial, we have completed on how to train a model for Sentence Detection in openNLP.