POS Tagger Example in Apache OpenNLP using Java

POS Tagger Example in Apache OpenNLP marks each word in a sentence with the word type.

**An Example:**

<table>
<thead>
<tr>
<th>Input to POS Tagger</th>
<th>John is 27 years old.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output of POS Tagger</td>
<td>John_NNP is_VBZ 27_CD years_NNS old_JJ_.</td>
</tr>
</tbody>
</table>

The word types are the tags attached to each word. These Parts Of Speech tags used are from Penn Treebank.

<table>
<thead>
<tr>
<th>NNP</th>
<th>Proper Noun, Singular</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBZ</td>
<td>Verb, 3rd person singular present</td>
</tr>
<tr>
<td>CD</td>
<td>Cardinal Number</td>
</tr>
<tr>
<td>NNS</td>
<td>Noun, Plural</td>
</tr>
<tr>
<td>JJ</td>
<td>Adjective</td>
</tr>
</tbody>
</table>

For a complete list of Parts Of Speech tags from Penn Treebank, please refer [https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html](https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html)

**Following are the steps to obtain the tags pragmatically in java using apache openNLP**

- **Step 1**: Tokenize the given input sentence into tokens.

  ```java
  String sentence = "John is 27 years old.";
  // tokenize the sentence
  tokenModelIn = new FileInputStream("en-token.bin");
  TokenizerModel tokenModel = new TokenizerModel(tokenModelIn);
  Tokenizer tokenizer = new TokenizerME(tokenModel);
  String tokens[] = tokenizer.tokenize(sentence);
  ```

- **Step 2**: Read the parts-of-speech maxent model, “en-pos-maxent.bin” into a stream.

  ```java
  InputStream posModelIn = new FileInputStream("en-pos-maxent.bin");
  ```

- **Step 3**: Read the stream into parts-of-speech model, POSModel.

  ```java
  POSModel posModel = new POSModel(posModelIn);
  ```

- **Step 4**: Load the model into parts-of-speech tagger, POSTaggerME

  ```java
  POSTaggerME posTagger = new POSTaggerME(posModel);
  ```
Step 5: Grab the tags using the method `POSTaggerME.tag()`, and probability for the tag to be given using the method `PosTaggerME.probs();`

```java
String tags[] = posTagger.tag(tokens);
double probs[] = posTagger.probs();
```

Step 6: Finally, print what we got, the token, their respective tags and probabilities of the tags.

The whole program at a glance is given below:

```java
/*
 * www.tutorialkart.com
 * POS Tagger Example in Apache OpenNLP using Java
 */
public class POSTaggerExample {

    public static void main(String[] args) {

        InputStream tokenModelIn = null;
        InputStream posModelIn = null;

        try {
            String sentence = "John is 27 years old.";
            // tokenize the sentence
            tokenModelIn = new FileInputStream("en-token.bin");
            TokenizerModel tokenModel = new TokenizerModel(tokenModelIn);
            Tokenizer tokenizer = new TokenizerME(tokenModel);
            String tokens[] = tokenizer.tokenize(sentence);

            // Parts-OF-Speech Tagging
            // reading parts-of-speech model to a stream
            posModelIn = new FileInputStream("en-pos-maxent.bin");
            // loading the parts-of-speech model from stream
            POSModel posModel = new POSModel(posModelIn);
            // initializing the parts-of-speech tagger with model
            POSTaggerME posTagger = new POSTaggerME(posModel);
            // Tagger tagging the tokens
            String tags[] = posTagger.tag(tokens);

            // Getting the probabilities of the tags given to the tokens
            double probs[] = posTagger.probs();

            System.out.println("Token\t\tTag\t\tProbability
---------------------------------------------");
            for(int i=0;i<tokens.length;i++){
                System.out.println(tokens[i] +"\t"+tags[i] +"\t"+probs[i]);
            }
        }
        catch (IOException e) {
            // Model loading failed, handle the error
            e.printStackTrace();
        }
        finally {
            if(tokenModelIn != null) {
```

```java
```
When the above program is run, the output to the console is shown below:

<table>
<thead>
<tr>
<th>Program Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token : Tag : Probability</td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td>John : NNP : 0.9874932809932121</td>
</tr>
<tr>
<td>is : VBZ : 0.9667574183085389</td>
</tr>
<tr>
<td>27 : CD : 0.9890000667325892</td>
</tr>
<tr>
<td>years : NNS : 0.97918326660035</td>
</tr>
<tr>
<td>old : JJ : 0.989475224172251</td>
</tr>
<tr>
<td>. . : . : 0.9923321017451305</td>
</tr>
</tbody>
</table>

The structure of the project is shown below:

Structure of the project

Please note that in this example, the model files, en-pos-maxent.bin and en-token.bin are placed right under the project folder. Please find the models at [http://opennlp.sourceforge.net/models-1.5/](http://opennlp.sourceforge.net/models-1.5/).

**Conclusion**:

In this Apache openNLP Tutorial, we have seen how to tag parts of speech to the words in a sentence using POSModel and POSTaggerME classes of openNLP Tagger API.

Following are some of the other example programs we have,

- Named Entity Extraction
- Sentence Detection
# Learn OpenNLP

- [ ] OpenNLP Tutorial
- [ ] Setup Java Project with OpenNLP in Eclipse
- [ ] OpenNLP Models

## Detection / Extraction using Java API

- [ ] Tokenizer Example
- [ ] Sentence Detection Example
- [ ] Parts-Of-Speech Tagger Example
- [ ] Chunker Example
- [ ] Lemmatizer Example
- [ ] Named Entity Extraction Example

## Training using Java API

- [ ] Sentence Detection Model Training
- [ ] Name Entity Finder Model Training
- [ ] Document Categorizer Training - Maximum Entropy
- [ ] Document Categorizer Training - Naive Bayes
- [ ] Document Categorizer with N-gram features used
- [ ] Language Detector Training Example

## Command Line Tools

- [ ] Setup and start using Command Line Tools

## Useful Resources

- [ ] How to Learn Programming