



OPENNLP

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

Apache **OpenNLP** is an open source Java library which is used process Natural Language text. OpenNLP provides services such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and co-reference resolution, etc.

In this tutorial, we will understand how to use the OpenNLP library to build an efficient text processing service.

Audience

This tutorial has been prepared for beginners to make them understand how to use the OpenNLP library, and thus help them in building text processing services using this library.

Prerequisites

For this tutorial, it is assumed that the readers have a prior knowledge of Java programming language.

Copyright & Disclaimer

© Copyright 2016 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial.....	i
Audience.....	i
Prerequisites.....	i
Copyright & Disclaimer.....	i
Table of Contents.....	ii
1. OPENNLP – OVERVIEW	1
What is Open NLP?	1
Features of OpenNLP.....	1
Open NLP Models.....	3
2. OPENNLP – ENVIRONMENT	5
Installing OpenNLP	5
Setting the Classpath.....	7
Eclipse Installation.....	9
3. OPENNLP – REFERENCED API.....	16
Sentence Detection	16
Tokenization.....	17
NameEntityRecognition	18
Finding the Parts of Speech.....	18
Parsing the Sentence.....	19
Chunking	20
4. OPEN NLP – SENTENCE DETECTION.....	21
Sentence Detection Using Java	21
Sentence Detection Using OpenNLP.....	22
Detecting the Positions of the Sentences.....	24
Sentences along with their Positions.....	27

Sentence Probability Detection	28
5. OPEN NLP – TOKENIZATION	30
Tokenizing using OpenNLP	30
Retrieving the Positions of the Tokens	36
Tokenizer Probability	41
6. OPEN NLP – NAMED ENTITY RECOGNITION	44
Named Entity Recognition using open NLP	44
Names along with their Positions	47
Finding the Names of the Location	48
NameFinder Probability	50
7. OPENNLP – FINDING PARTS OF SPEECH	52
Tagging the Parts of Speech	52
POS Tagger Performance	55
POS Tagger Probability	56
8. OPENNLP – PARSING THE SENTENCES	59
Parsing Raw Text using OpenNLP Library	59
9. OPENNLP – CHUNKING SENTENCES	62
Chunking a Sentence using OpenNLP	62
Detecting the Positions of the Tokens	65
Chunker Probability Detection	67
10. OPENNLP – COMMAND LINE INTERFACE	70
Tokenizing	70
Sentence Detection	71
Named Entity Recognition	71
Parts of Speech Tagging	72

1. OpenNLP – Overview

NLP is a set of tools used to derive meaningful and useful information from natural language sources such as web pages and text documents.

What is Open NLP?

Apache **OpenNLP** is an open-source Java library which is used to process natural language text. You can build an efficient text processing service using this library.

OpenNLP provides services such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and co-reference resolution, etc.

A Brief History of OpenNLP

- In 2010, OpenNLP entered the Apache incubation.
- In 2011, Apache OpenNLP 1.5.2 Incubating was released, and in the same year, it graduated as a top-level Apache project.
- In 2015, OpenNLP was 1.6.0 released.

Features of OpenNLP

Following are the notable features of OpenNLP –

- **Named Entity Recognition (NER):** Open NLP supports NER, using which you can extract names of locations, people and things even while processing queries.
- **Summarize:** Using the **summarize** feature, you can summarize Paragraphs, articles, documents or their collection in NLP.
- **Searching:** In OpenNLP, a given search string or its synonyms can be identified in given text, even though the given word is altered or misspelled.
- **Tagging (POS):** Tagging in NLP is used to divide the text into various grammatical elements for further analysis.
- **Translation:** In NLP, Translation helps in translating one language into another.
- **Information grouping:** This option in NLP groups the textual information in the content of the document, just like Parts of speech.
- **Natural Language Generation:** It is used for generating information from a database and automating the information reports such as weather analysis or medical reports.

- **Feedback Analysis:** As the name implies, various types of feedbacks from people are collected, regarding the products, by NLP to analyze how well the product is successful in winning their hearts.
- **Speech recognition:** Though it is difficult to analyze human speech, NLP has some built-in features for this requirement.

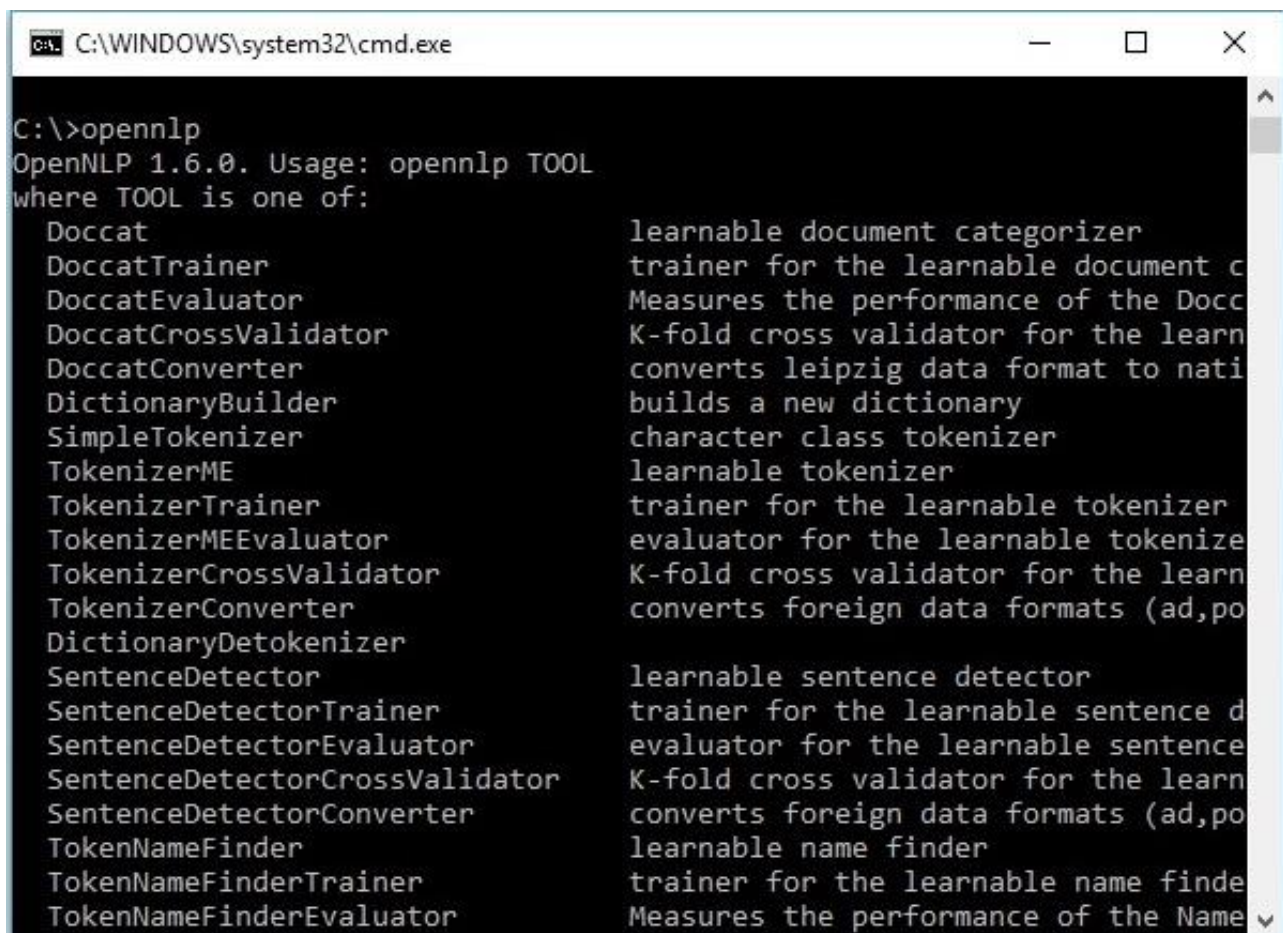
Open NLP API

The Apache OpenNLP library provides classes and interfaces to perform various tasks of natural language processing such as sentence detection, tokenization, finding a name, tagging the parts of speech, chunking a sentence, parsing, co-reference resolution, and document categorization.

In addition to these tasks, we can also train and evaluate our own models for any of these tasks.

OpenNLP CLI

In addition to the library, OpenNLP also provides a Command Line Interface (CLI), where we can train and evaluate models. We will discuss this topic in detail in the last chapter of this tutorial.



```

C:\WINDOWS\system32\cmd.exe
C:\>opennlp
OpenNLP 1.6.0. Usage: opennlp TOOL
where TOOL is one of:
  Doccat                learnable document categorizer
  DoccatTrainer          trainer for the learnable document c
  DoccatEvaluator        Measures the performance of the Docc
  DoccatCrossValidator  K-fold cross validator for the learn
  DoccatConverter        converts leipzig data format to nati
  DictionaryBuilder      builds a new dictionary
  SimpleTokenizer        character class tokenizer
  TokenizerME            learnable tokenizer
  TokenizerTrainer       trainer for the learnable tokenizer
  TokenizerMEEvaluator   evaluator for the learnable tokenize
  TokenizerCrossValidator K-fold cross validator for the learn
  TokenizerConverter     converts foreign data formats (ad,po
  DictionaryDetokenizer
  SentenceDetector        learnable sentence detector
  SentenceDetectorTrainer trainer for the learnable sentence d
  SentenceDetectorEvaluator evaluator for the learnable sentence
  SentenceDetectorCrossValidator K-fold cross validator for the learn
  SentenceDetectorConverter converts foreign data formats (ad,po
  TokenNameFinder        learnable name finder
  TokenNameFinderTrainer trainer for the learnable name finde
  TokenNameFinderEvaluator Measures the performance of the Name

```

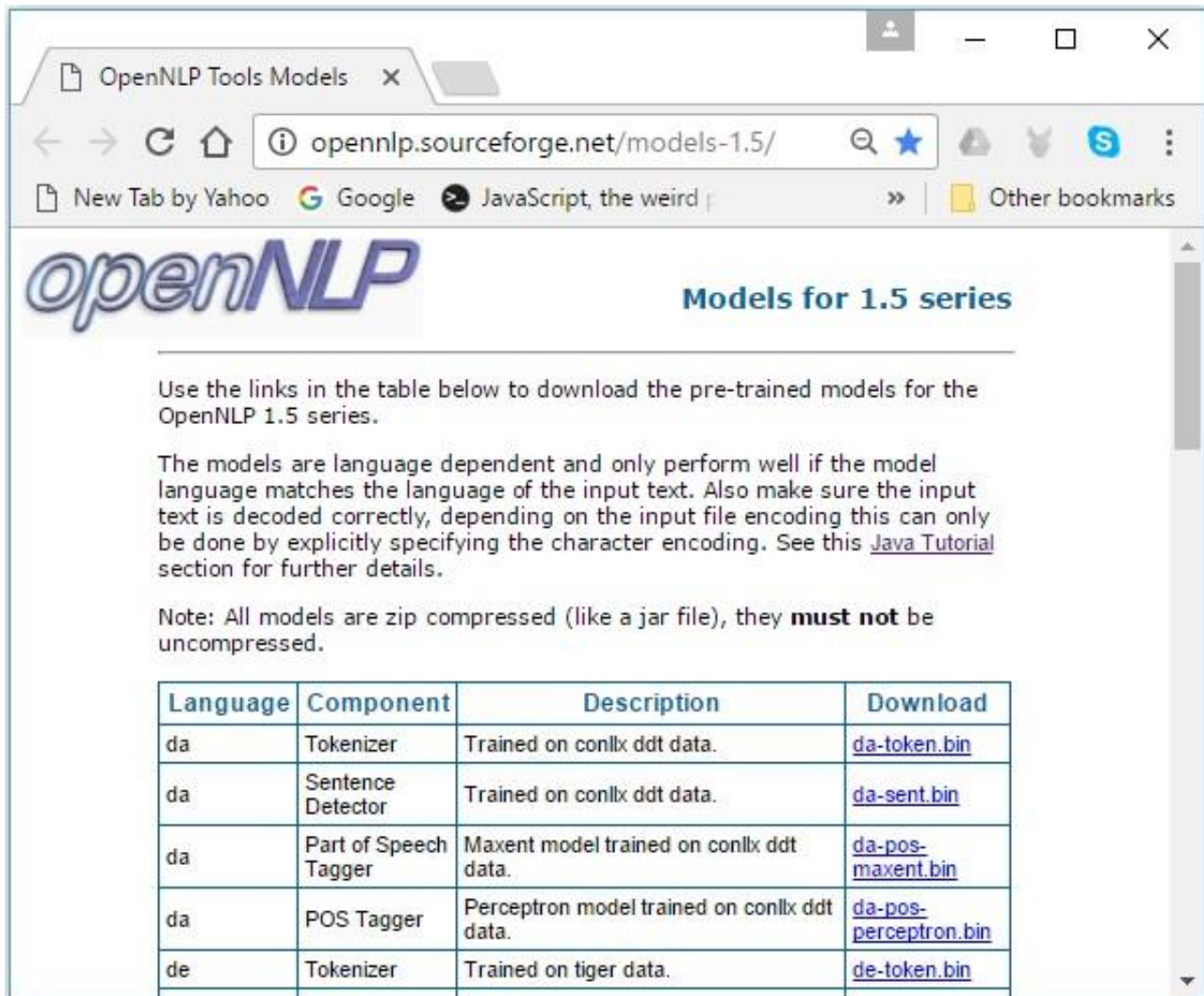
Open NLP Models

To perform various NLP tasks, OpenNLP provides a set of predefined models. This set includes models for different languages.

Downloading the models

You can follow the steps given below to download the predefined models provided by OpenNLP.

Step 1: Open the index page of OpenNLP models by clicking the following link: <http://opennlp.sourceforge.net/models-1.5/>



Use the links in the table below to download the pre-trained models for the OpenNLP 1.5 series.

The models are language dependent and only perform well if the model language matches the language of the input text. Also make sure the input text is decoded correctly, depending on the input file encoding this can only be done by explicitly specifying the character encoding. See this [Java Tutorial](#) section for further details.

Note: All models are zip compressed (like a jar file), they **must not** be uncompressed.

Language	Component	Description	Download
da	Tokenizer	Trained on conlx ddt data.	da-token.bin
da	Sentence Detector	Trained on conlx ddt data.	da-sent.bin
da	Part of Speech Tagger	Maxent model trained on conlx ddt data.	da-pos-maxent.bin
da	POS Tagger	Perceptron model trained on conlx ddt data.	da-pos-perceptron.bin
de	Tokenizer	Trained on tiger data.	de-token.bin

Step 2: On visiting the given link, you will get to see a list of components of various languages and the links to download them. Here, you can get the list of all the predefined models provided by OpenNLP.

en	Tokenizer	Trained on opennlp training data.	en-token.bin
en	Sentence Detector	Trained on opennlp training data.	en-sent.bin
en	POS Tagger	Maxent model with tag dictionary.	en-pos-maxent.bin
en	POS Tagger	Perceptron model with tag dictionary.	en-pos-perceptron.bin
en	Name Finder	Date name finder model.	en-ner-date.bin
en	Name Finder	Location name finder model.	en-ner-location.bin
en	Name Finder	Money name finder model.	en-ner-money.bin
en	Name Finder	Organization name finder model.	en-ner-organization.bin
en	Name Finder	Percentage name finder model.	en-ner-percentage.bin
en	Name Finder	Person name finder model.	en-ner-person.bin
en	Name Finder	Time name finder model.	en-ner-time.bin
en	Chunker	Trained on conll2000 shared task data.	en-chunker.bin
en	Parser		en-parser-chunking.bin
en	Coreference		coref

Download all these models to the folder **C:/OpenNLP_models/**, by clicking on their respective links. All these models are language dependent and while using these, you have to make sure that the model language matches with the language of the input text.

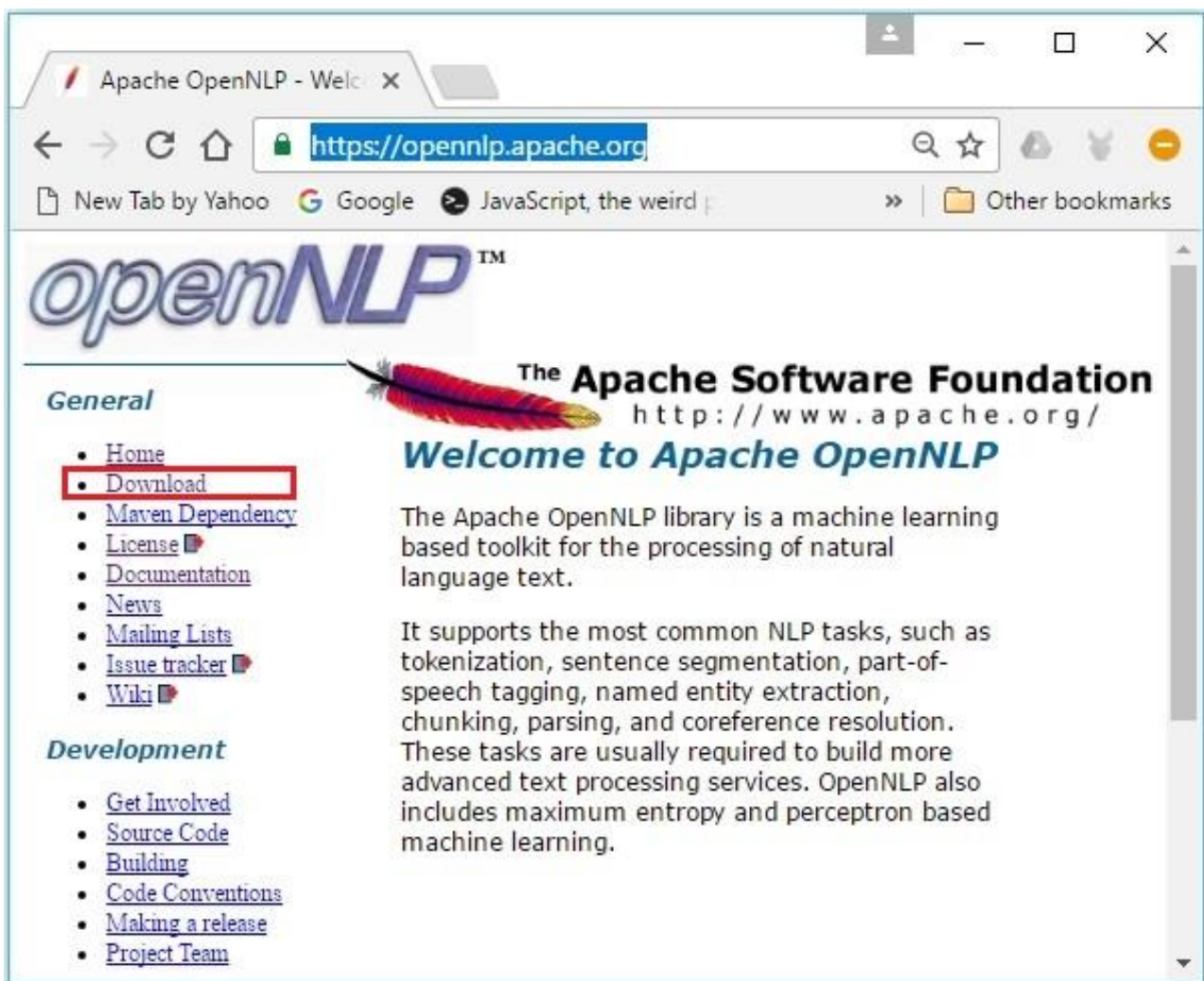
2. OpenNLP – Environment

In this chapter, we will discuss how you can setup OpenNLP environment in your system. Let's start with the installation process.

Installing OpenNLP

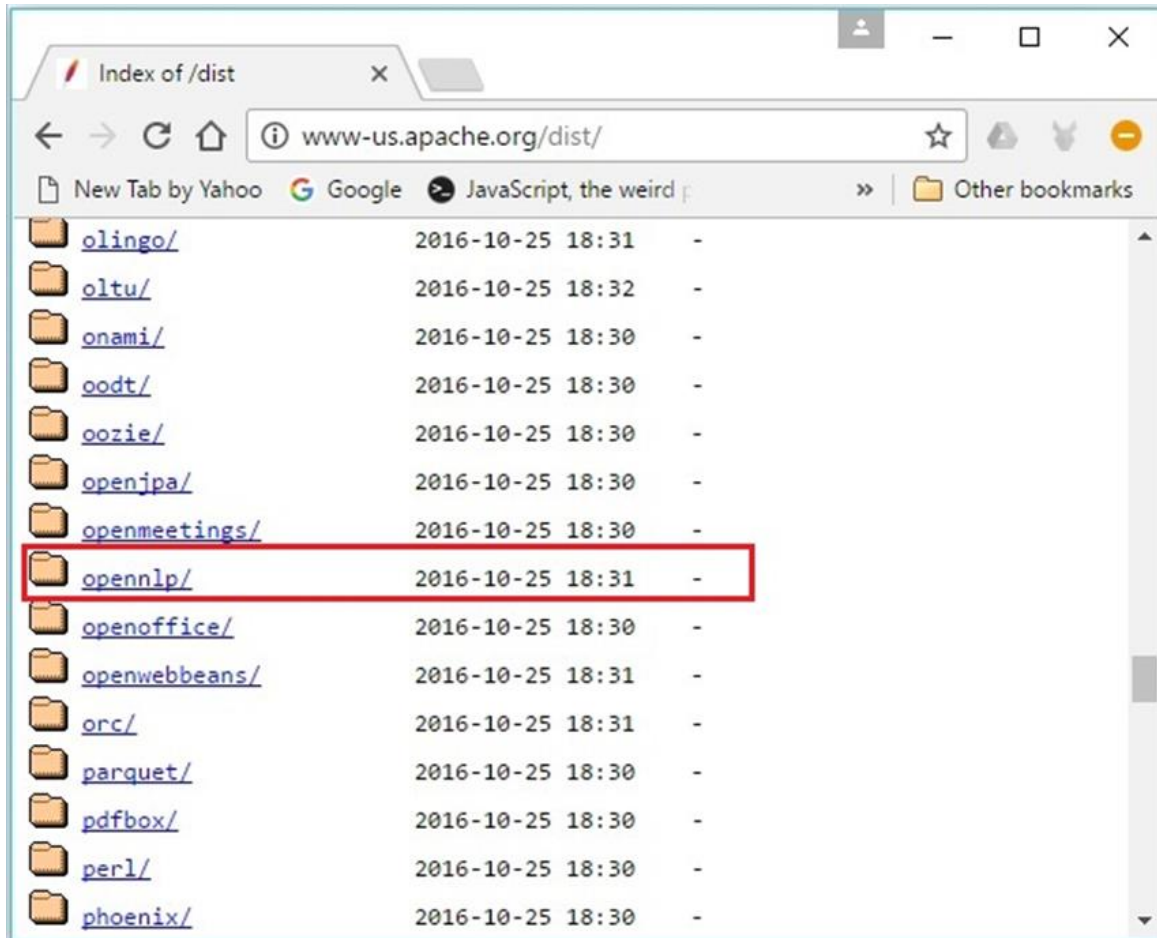
Following are the steps to download **Apache OpenNLP library** in your system.

Step 1: Open the homepage of **Apache OpenNLP** by clicking the following link: <https://opennlp.apache.org/>

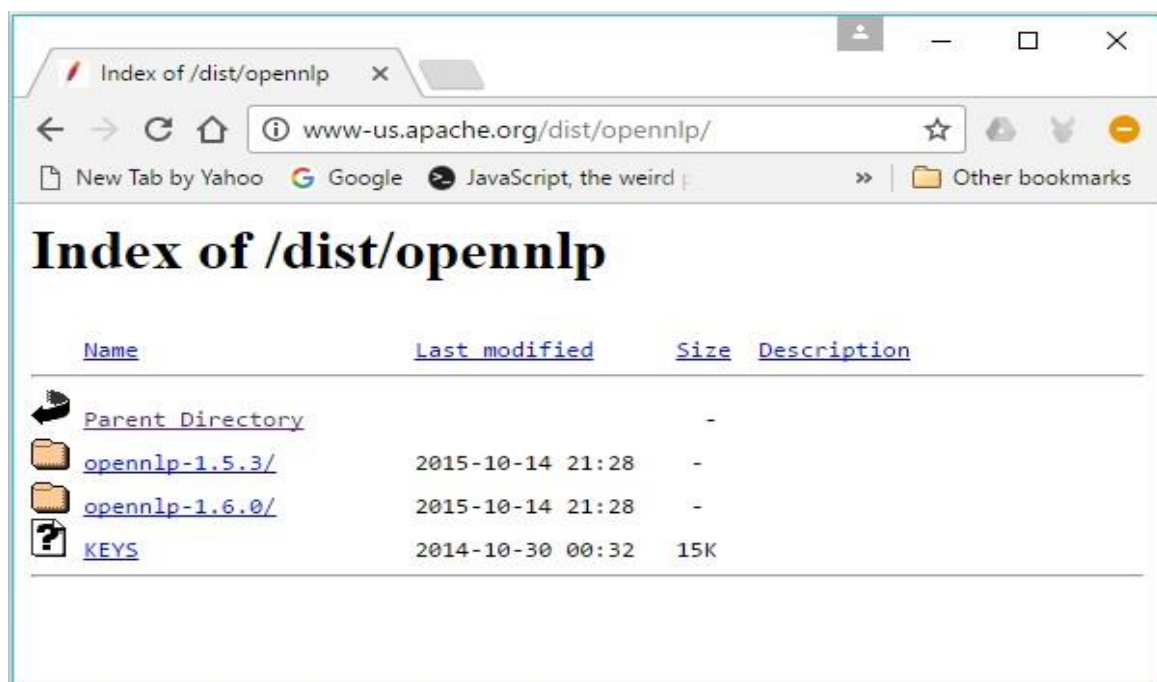


Step 2: Now, click on the **Downloads** link. On clicking, you will be directed to a page where you can find various mirrors which will redirect you to the Apache Software Foundation Distribution directory.

Step 3: In this page you can find links to download various Apache distributions. Browse through them and find the OpenNLP distribution and click it.

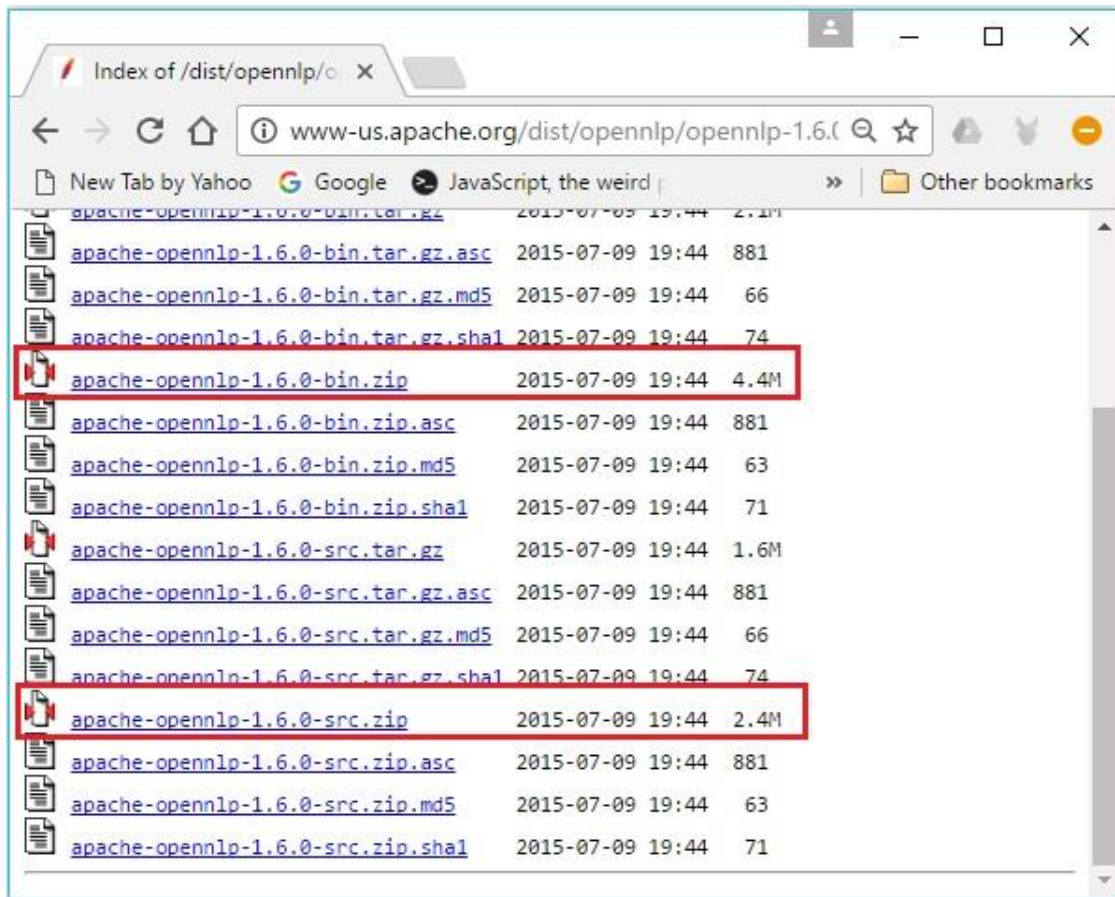


Step 4: On clicking, you will be redirected to the directory where you can see the index of the OpenNLP distribution, as shown below.



Click on the latest version from the available distributions.

Step 5: Each distribution provides Source and Binary files of OpenNLP library in various formats. Download the source and binary files, **apache-opennlp-1.6.0-bin.zip** and **apache-opennlp-1.6.0-src.zip** (for Windows).



Setting the Classpath

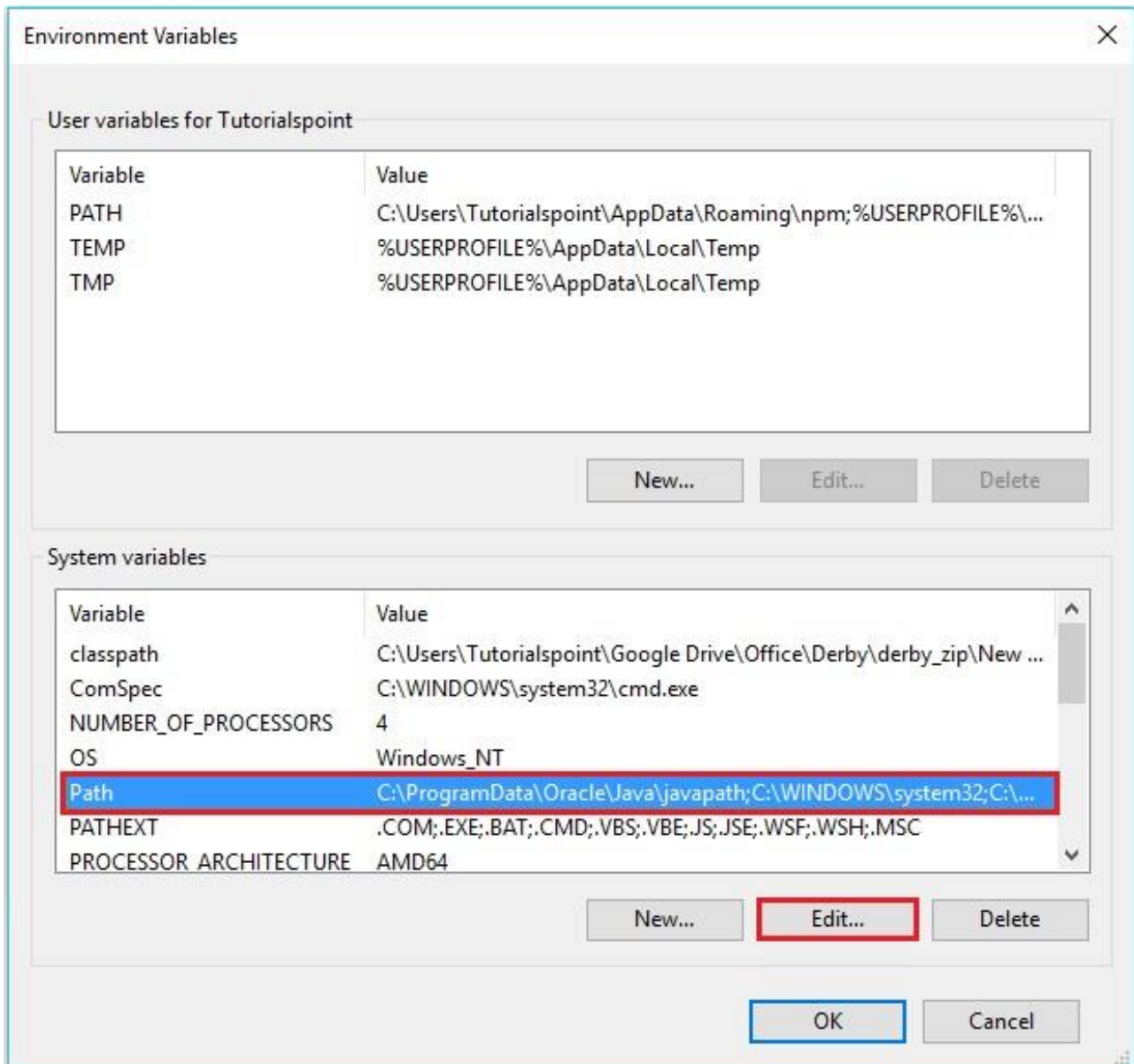
After downloading the OpenNLP library, you need to set its path to the **bin** directory. Assume that you have downloaded the OpenNLP library to the E drive of your system.

Now, follow the steps that are given below:

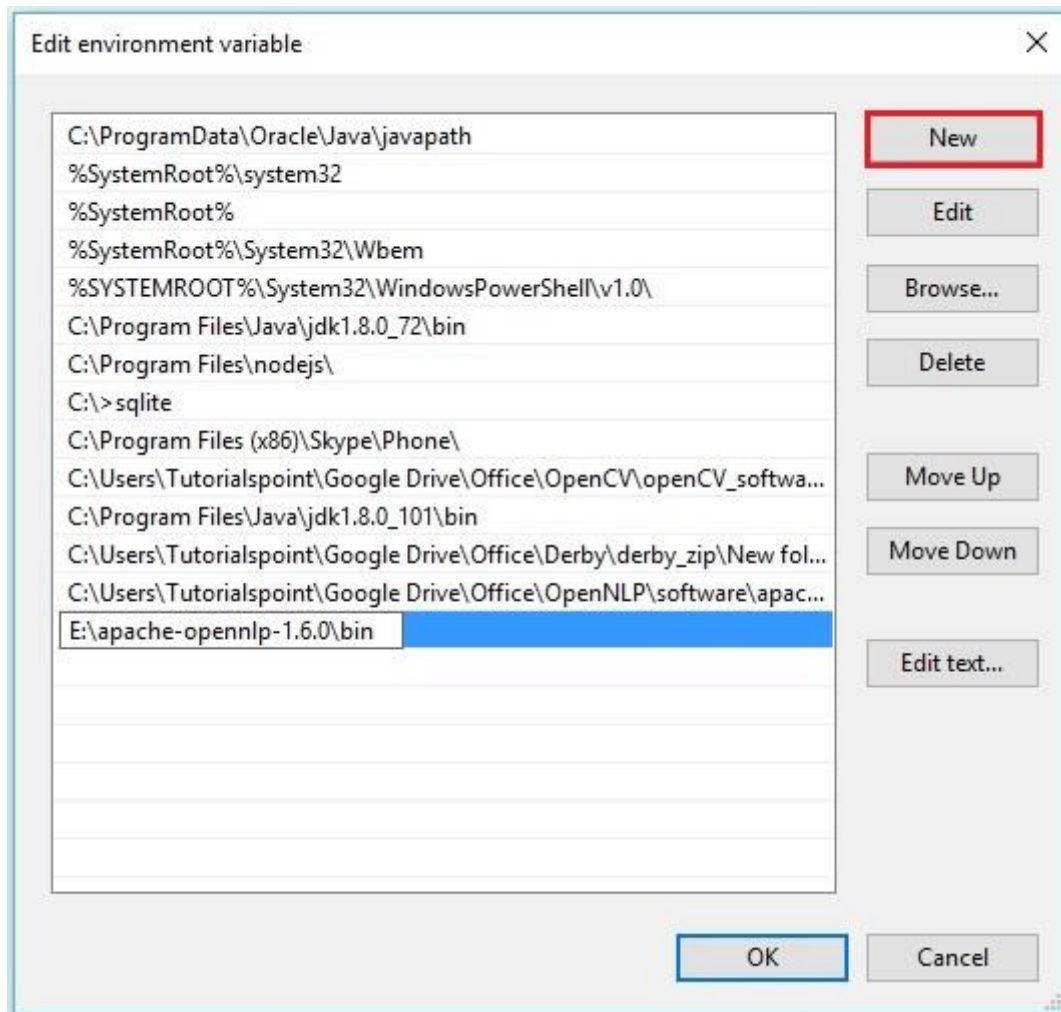
Step 1: Right-click on 'My Computer' and select 'Properties'.

Step 2: Click on the 'Environment Variables' button under the 'Advanced' tab.

Step 3: Select the **path** variable and click the **Edit** button, as shown in the following screenshot.



Step 4: In the *Edit Environment Variable* window, click the **New** button and add the path for OpenNLP directory **E:\apache-opennlp-1.6.0\bin** and click the **OK** button, as shown in the following screenshot.



Eclipse Installation

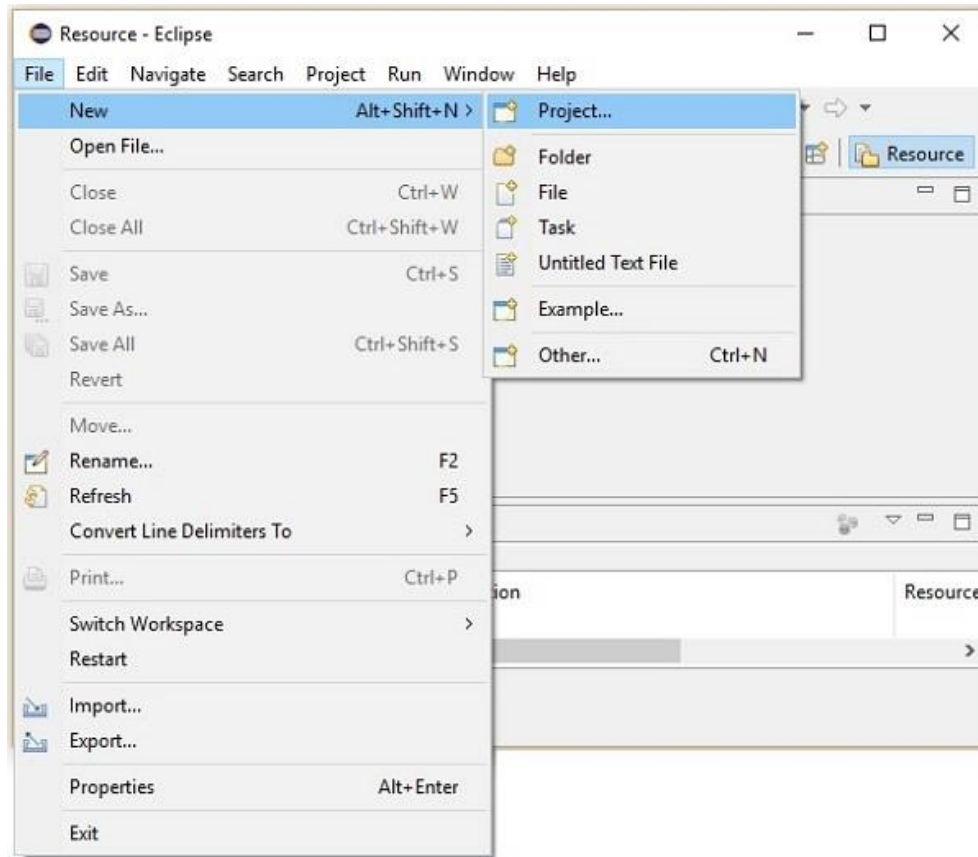
You can set the Eclipse environment for OpenNLP library, either by setting the **Build path** to the JAR files or by using **pom.xml**.

Setting Build Path to the JAR Files

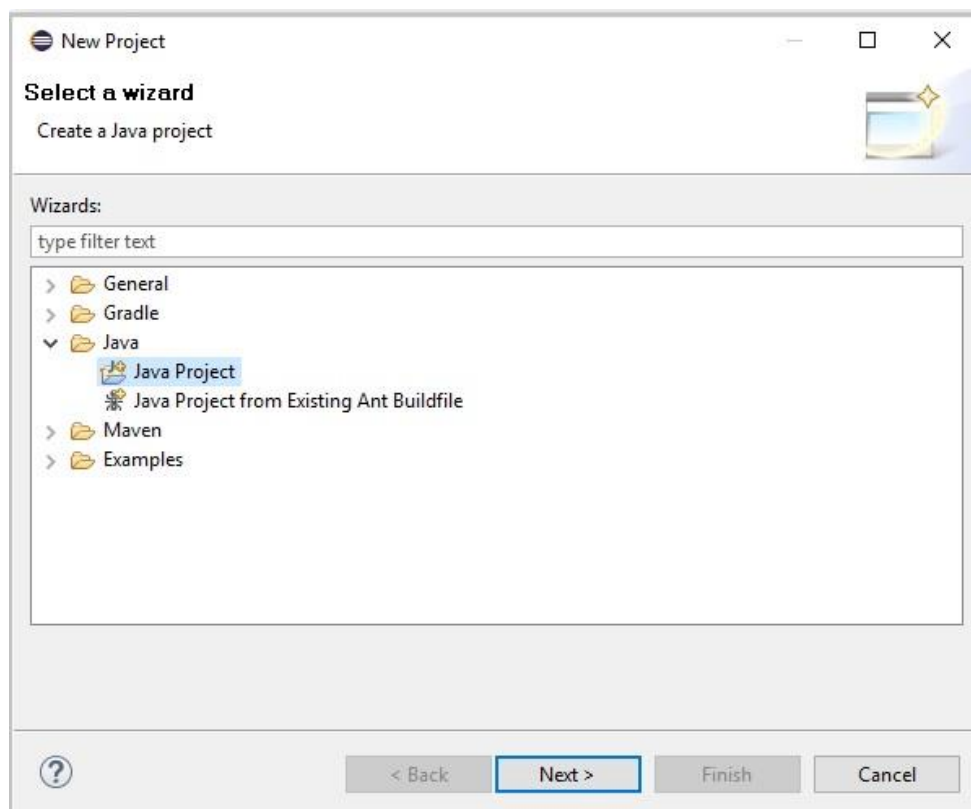
Follow the steps given below to install OpenNLP in Eclipse:

Step 1: Make sure that you have Eclipse environment installed in your system.

Step 2: Open Eclipse. Click File -> New -> Open a new project, as shown below.



Step 3: You will get the **New Project** wizard. In this wizard, select Java project and proceed by clicking the **Next** button.



Step 4: Next, you will get the **New Java Project wizard**. Here, you need to create a new project and click the **Next** button, as shown below.

New Java Project

Create a Java Project
Create a Java project in the workspace or in an external location.

Project name:

Use default location

Location: [Browse...](#)

JRE

Use an execution environment JRE:

Use a project specific JRE:

Use default JRE (currently 'jre1.8.0_72') [Configure JREs...](#)

Project layout

Use project folder as root for sources and class files

Create separate folders for sources and class files [Configure default...](#)

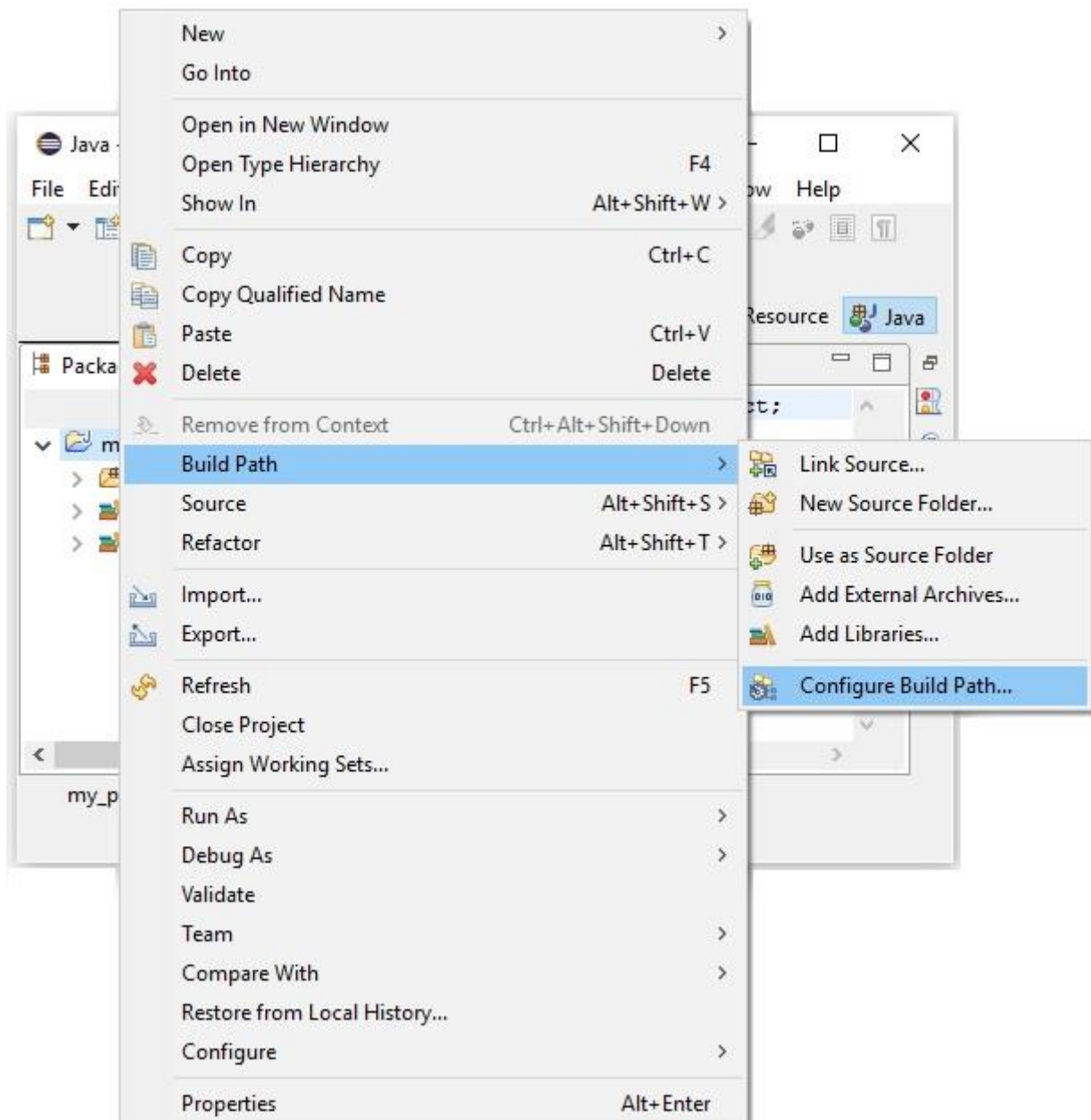
Working sets

Add project to working sets

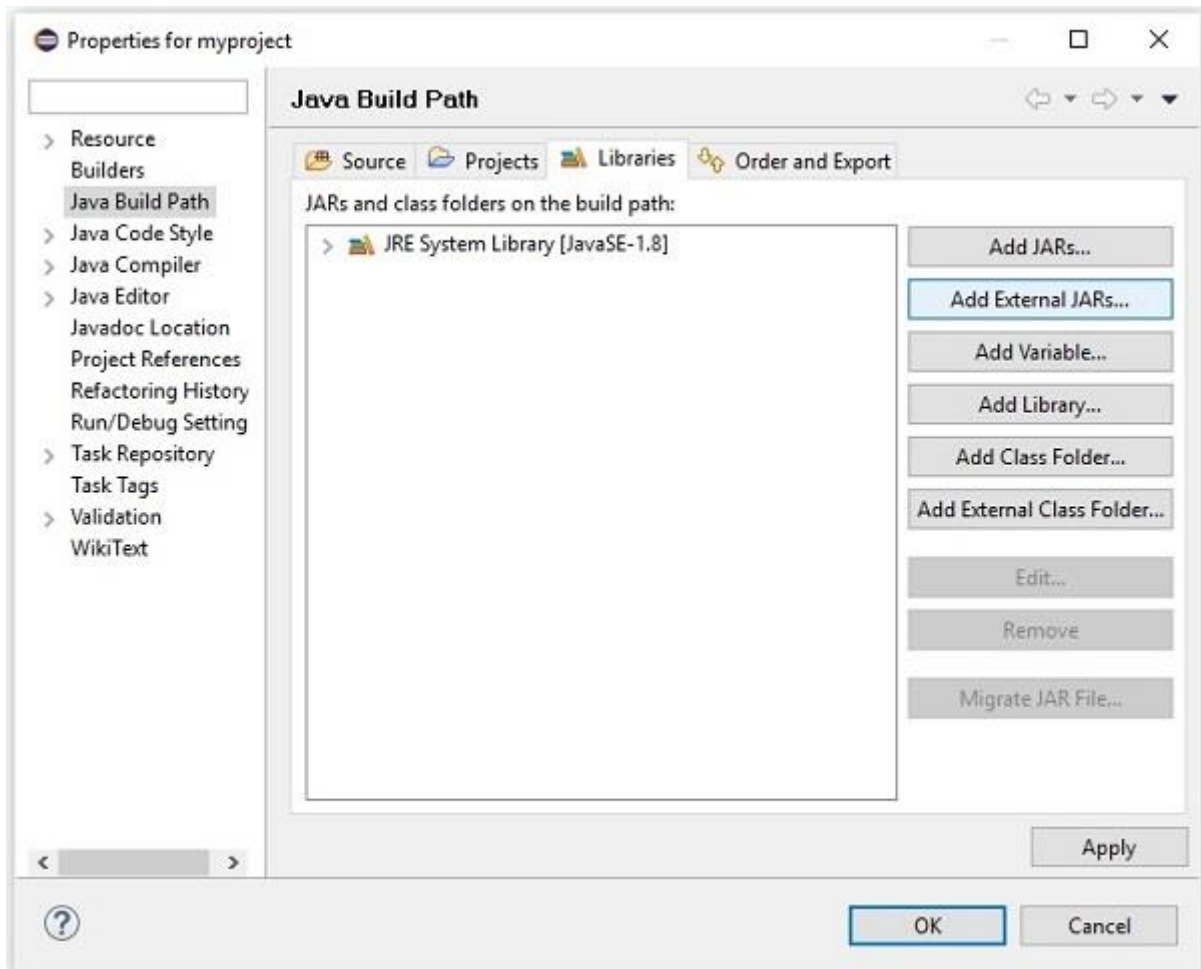
Working sets: [Select...](#)

[? < Back](#) [Next >](#) [Finish](#) [Cancel](#)

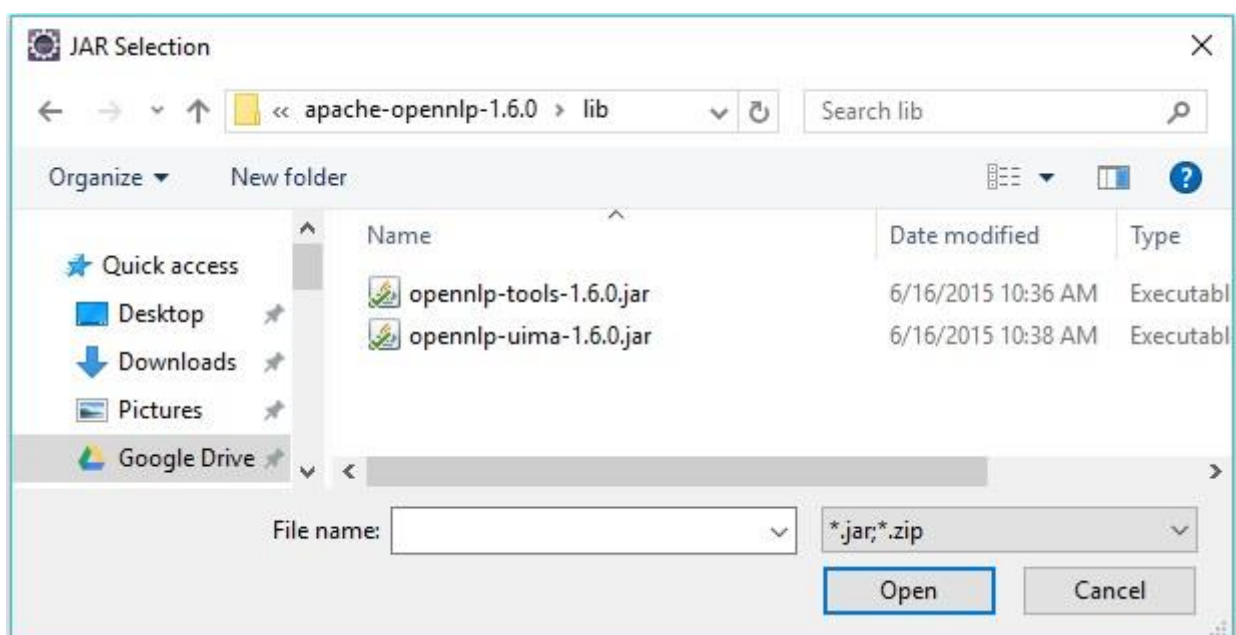
Step 5: After creating a new project, right-click on it, select **Build Path** and click **Configure Build Path**.



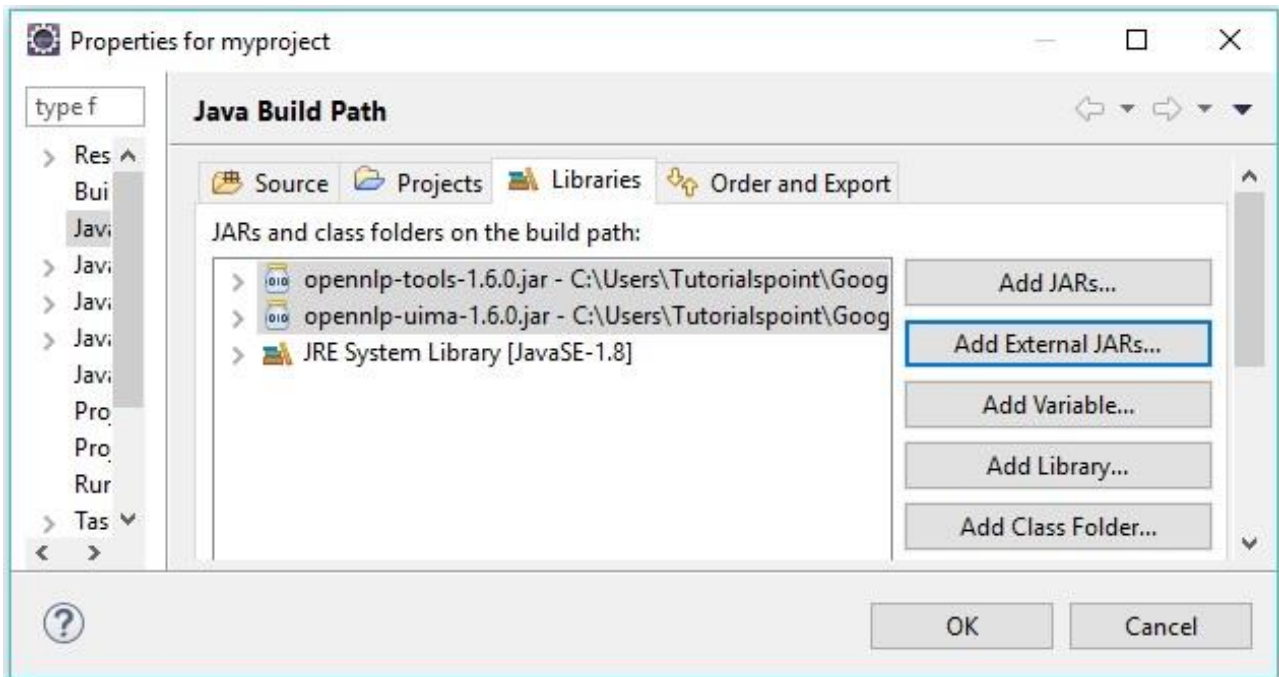
Step 6: Next, you will get the **Java Build Path** wizard. Here, click the **Add External JARs** button, as shown below.



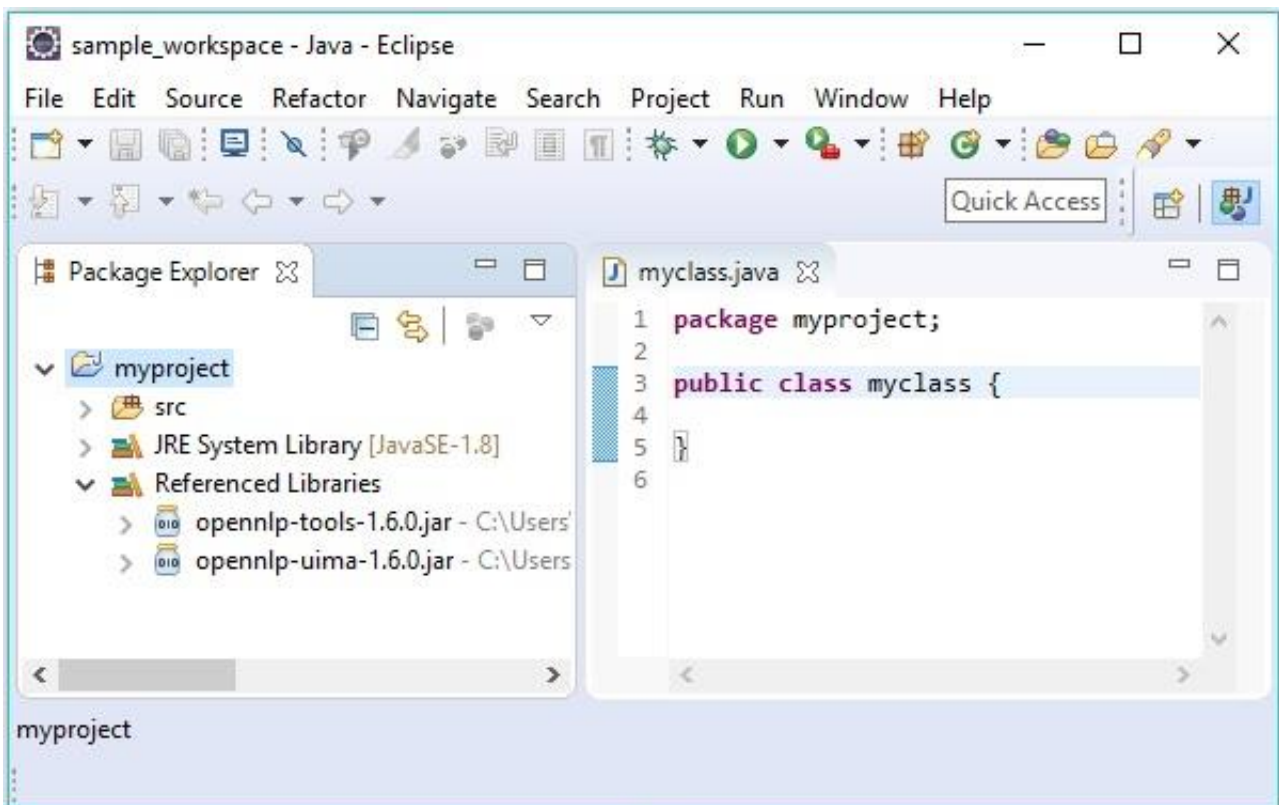
Step 7: Select the jar files **opennlp-tools-1.6.0.jar** and **opennlp-uima-1.6.0.jar** located in the **lib** folder of **apache-opennlp-1.6.0** folder.



On clicking the **Open** button in the above screen, the selected files will be added to your library.



On clicking **OK**, you will successfully add the required JAR files to the current project and you can verify these added libraries by expanding the Referenced Libraries, as shown below.



Using pom.xml

Convert the project into a Maven project and add the following code to its **pom.xml**.

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>myproject</groupId>
  <artifactId>myproject</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <build>
    <sourceDirectory>src</sourceDirectory>
    <plugins>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.5.1</version>
        <configuration>
          <source>1.8</source>
          <target>1.8</target>
        </configuration>
      </plugin>
    </plugins>
  </build>
  <dependencies>
    <dependency>
      <groupId>org.apache.opennlp</groupId>
      <artifactId>opennlp-tools</artifactId>
      <version>1.6.0</version>
    </dependency>
    <dependency>
      <groupId>org.apache.opennlp</groupId>
      <artifactId>opennlp-uima</artifactId>
      <version>1.6.0</version>
    </dependency>
  </dependencies>
</project>
```

End of ebook preview

If you liked what you saw...

Buy it from our store @ <https://store.tutorialspoint.com>