# Apache OpenNLP Tutorial

Apache OpenNLP is an open source project that is cross platform and written inJava. It is a toolkit, for NLP(Natural Language Processing), based on machine learning.

In this Apache OpenNLP Tutorial, we shall learn the tools it provides to solve some of theNatural Language Processing tasks like Named Entity Recognition, Sentence Detection, Chunking, Tokenization, Parts-of-Speech Tagging, Document Classification or Categorization through Java API and Command Line Interface.

## Prerequisites

To understand the usage of Command Line Interface of Apache OpenNLP, no programming skill is required. Basic understanding of Natural Language Processing tasks, Machine Learning parameters would suffice.

To understand the usage of Apache OpenNLP's Java API, basic Java Programming skills is required along with a little idea on Natural Language Processing tasks and little idea of Machine Learning parameters like number of epochs and cut-off. Appropriate intuition would be provided in the corresponding tutorials for Natural Language Processing tasks.

## What is Natural Language Processing and the tasks it deals with

Natural Language Processing is all about the interaction between computer and human. Generally, humans interact with each other using vocabulary. And the language they are using (say English, Spanish, Hindi, etc.,) has some set of rules. It does not happen all the time that all people speaking these languages to communicate use the grammar of the language alike. Different people might use different words for conveying the same information. But as people around them have already known them or used to such kind, can understand or get the summary or inference from what they are saying.

Humans perceive information like context, inference etc., from the sentences formed using vocabulary and grammar. And when a machine or computer is expected to understand the context, inference or summary or useful information from the data it gets from a human, there are some gaps that needs to be filled. These gaps are the tasks that Natural Language Processing deals with, to make a machine understand a human language or speak to human in natural language.

Apache OpenNLP is an open-source library that provides solutions to some of the Natural Language Processing tasks through its APIs and command line tools. Apache OpenNLP uses machine learning approach for the tasks of processing natural language. It also provides some of the pre-built models for some of the tasks. Following are the tasks to which Apache OpenNLP provides APIs[http://opennlp.apache.org/docs/1.7.2/manual/opennlp.html], and those we deal with examples in this OpenNLP Tutorial :

**Note** : To setup a Java Project with Eclipse, refer how to setup OpenNLP in Java with Eclipse.

# Apache OpenNLP Tutorial – APIs

## Named Entity Recognition (NER)

Named Entity Recognition is to find named entities like person, place, organisation or a thing in a given sentence. OpenNLP has built models for NER which can be directly used and also helps in training a model for the custom datat we have. Named Entity Recognition Example with existing model Named Entity Recognition (NER) **Training** Example

## Document Categorizer

Categorizing or Classifying a given document to one of the pre-defined categories is what a Document Categorizer does. OpenNLP provides an API that helps in categorizing or classifying documents. As categorizing documents cannot be generalized like NER, there are no pre-built models available, but anyone can build a model by his/her own requirements. Document classification using Maximum Entropy (Maxent) Document classification using Naive Bayes Example to demonstrate the usage of NGram feature for document classification.

## Sentence Detection

The process of identifying sentences in a paragraph or a document or a text file is called Sentence Detection. OpenNLP supports Sentence Detection through its API. It provide pre-built models for sentence detection, and also a means to build a model for requirement specific data. Sentence Detection Example in Apache OpenNLP using Java Sentence Detection **Training** Example in Apache OpenNLP using Java

## Parts of Speech Tagging

Understanding grammar is an important task in NLP. Identifying Parts of Speech in a given sentence is a stepping block to understand grammar. Apache OpenNLP provides APIs to train a model that can identify Parts of Speech or use a pre-built model and identify Parts of Speech in a sentence. Parts of Speech Tagger Example in Apache OpenNLP using Java

## Tokenization

Tokenization is a process of breaking down the given sentence into smaller pieces like words, punctuation marks, numbers etc. Apache OpenNLP provides APIs to train a model or use a pre-built model and break a sentence into smaller pieces. Tokenizer Example in Apache OpenNLP using Java

## Lemmatization

Lemmatization is a process of removing any changes in form of the word like tense, gender, mood, etc. and return dictionary or base form of the word. Lemmatization Example

## Language Detection

Language Detection is a task of finding the natural language to which the sample text provided belongs to. Language Detection Example

## Command Line Interface of Apache OpenNLP

All the tools included in OpenNLP could be accessed through command line interface. Following are some of the examples :

- Usage of Apache OpenNLP's Command Line Interface.

## Conclusion

With this Apache OpenNLP tutorial we understood the overview of OpenNLP and the APIs it provides. Lets start getting hands on OpenNLP by setting up a Java Project with OpenNLP in Eclipseand trying out the APIs that it provides.

## 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