FREE eBook

# LEARNING

# nlp

Free unaffiliated eBook created from
**Stack Overflow contributors.**

#nlp

# Table of Contents

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: nlp

It is an unofficial and free nlp ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official nlp.



Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

# Chapter 1: Getting started with nlp

## Remarks

This section provides an overview of what nlp is, and why a developer might want to use it.

It should also mention any large subjects within nlp, and link out to the related topics. Since the Documentation for nlp is new, you may need to create initial versions of those related topics.

## Examples

### Stanford CoreNLP

Stanford CoreNLP is a popular Natural Language Processing toolkit supporting many core NLP tasks.

To download and install the program, either download a release package and include the necessary `*.jar` files in your classpath, or add the dependency off of Maven central. See the download page for more detail. For example:

```
curl http://nlp.stanford.edu/software/stanford-corenlp-full-2015-12-09.zip -o corenlp.zip
unzip corenlp.zip
cd corenlp
export CLASSPATH="$CLASSPATH:`pwd`/*
```

There are three supported ways to run the CoreNLP tools: (1) using the base fully customizable API, (2) using the Simple CoreNLP API, or (3) using the CoreNLP server. A simple usage example for each is given below. As a motivating use case, these examples will be for predicting the syntactic parse of a sentence.

1. **CoreNLP API**

```
public class CoreNLPDemo {
  public static void main(String[] args) {

    // 1. Set up a CoreNLP pipeline. This should be done once per type of annotation,
    //    as it's fairly slow to initialize.
    // creates a StanfordCoreNLP object, with POS tagging, lemmatization, NER, parsing,
and coreference resolution
    Properties props = new Properties();
    props.setProperty("annotators", "tokenize, ssplit, parse");
    StanfordCoreNLP pipeline = new StanfordCoreNLP(props);

    // 2. Run the pipeline on some text.
    // read some text in the text variable
    String text = "the quick brown fox jumped over the lazy dog"; // Add your text here!
    // create an empty Annotation just with the given text
    Annotation document = new Annotation(text);
    // run all Annotators on this text
    pipeline.annotate(document);
```

```
    // 3. Read off the result
    // Get the list of sentences in the document
    List<CoreMap> sentences = document.get(CoreAnnotations.SentencesAnnotation.class);
    for (CoreMap sentence : sentences) {
      // Get the parse tree for each sentence
      Tree parseTree = sentence.get(TreeAnnotations.TreeAnnotation.class);
      // Do something interesting with the parse tree!
      System.out.println(parseTree);
    }

  }
}
```

2. **Simple CoreNLP**

```
public class CoreNLPDemo {
  public static void main(String[] args) {
    String text = "The quick brown fox jumped over the lazy dog");  // your text here!
    Document document = new Document(text);  // implicitly runs tokenizer
    for (Sentence sentence : document.sentences()) {
      Tree parseTree = sentence.parse();  // implicitly runs parser
      // Do something with your parse tree!
      System.out.println(parseTree);
    }
  }
}
```

3. **CoreNLP Server**

   Start the server with the following (setting your classpath appropriately):

```
java -mx4g -cp "*" edu.stanford.nlp.pipeline.StanfordCoreNLPServer [port] [timeout]
```

   Get a JSON-formatted output for a given set of annotators, and print it to standard out:

```
 wget --post-data 'The quick brown fox jumped over the lazy dog.'
'localhost:9000/?properties={"annotators":"tokenize,ssplit,parse","outputFormat":"json"}'
-O -
```

   To get our parse tree from the JSON, we can navigate the JSON to `sentences[i].parse`.

Read Getting started with nlp online: https://riptutorial.com/nlp/topic/2613/getting-started-with-nlp

# Chapter 2: N-GRAMS

## Introduction

N-GRAMs are statistical models that predict the next word in the sentence by using the previous n-1 words. This type of statistical models that uses word sequences are also called Language Models. For instance we have a sentence "I can't read without my reading _____", we can tell that the next most likely word would be "glasses". N-GRAMS predicts the next word in the sequence by using the conditional probability of the next word. N-GRAM model is very essential in speech and language processing.

## Syntax

- The conditional probability of the next most likely word can be obtained by using a big corpus(Managed Collection of text or speech data), it is all bout counting things(words) from the corpus. The goal is to find P(w|h), which the probability of next word in the sequence given some history h.
- The Concept of the N-GRAM model is that instead of computing the probability of a word given its entire history, it shortens the history to previous few words. When we use only a single previous word to predict the next word it is called a Bi-GRAM model. For Example, we have P(glasses|reading), the probability of the word "glasses" given the previous word "reading" is computed as:(Refer to the example)

## Remarks

N-GRAM models are very important when we have to identify words in a noisy and ambiguous input. N-GRAM models are used in:

- Speech Recognition
- Hand Writing Recognition
- Spell Correction
- Machine Translation
- many other applications

You can read more about N-GRAM models in:

- Speech and Language Processing Book by Daniel Jurafsky and James H. Martin

## Examples

### Computing the Conditional Probability

P( *glasses* | *reading* ) = Count( *reading glasses* ) / Count( *reading* )

We count the sequences `reading glasses` and `glasses` from corpus and compute the probability.

Read N-GRAMS online: https://riptutorial.com/nlp/topic/8851/n-grams

# Chapter 3: OpenNLP

## Syntax

- opennlp SentenceDetector ./en-sent.bin < ./input.txt > output.txt
- Initialize SentenceDetectorME like this: SentenceDetectorME sentenceDetector = new SentenceDetectorME(model);
- Use ‘sentDetect’ method to get sentences like this: String sentences[] = sentenceDetector.sentDetect("string of information");

## Remarks

download models(like en-sent.bin) from the following link

## Examples

### Sentence Detection using openNLP using CLI and Java API

***using CLI:***

```
$ opennlp SentenceDetector ./en-sent.bin < ./input.txt > output.txt
```

***using API:***

```
import static java.nio.file.Files.readAllBytes;
import static java.nio.file.Paths.get;

import java.io.IOException;
import java.util.Objects;

public class FileUtils {
/**
 * Get file data as string
 *
 * @param fileName
 * @return
 */
    public static String getFileDataAsString(String fileName) {
        Objects.nonNull(fileName);
        try {
            String data = new String(readAllBytes(get(fileName)));
            return data;
        } catch (IOException e) {
            System.out.println(e.getMessage());
            return null;
        }
    }
}
```

class sentecedetectorutil:

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.util.Objects;

import opennlp.tools.sentdetect.SentenceDetectorME;
import opennlp.tools.sentdetect.SentenceModel;

public class SentenceDetectorUtil {
    private SentenceModel model = null;
    SentenceDetectorME sentenceDetector = null;

    public SentenceDetectorUtil(String modelFile) {
        Objects.nonNull(modelFile);
        initSentenceModel(modelFile);
        initSentenceDetectorME();
    }

    private void initSentenceDetectorME() {
        sentenceDetector = new SentenceDetectorME(model);
    }

    private SentenceModel initSentenceModel(String file) {
        InputStream modelIn;
        try {
            modelIn = new FileInputStream(file);
        } catch (FileNotFoundException e) {
            System.out.println(e.getMessage());
            return null;
        }

        try {
            model = new SentenceModel(modelIn);
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (modelIn != null) {
                try {
                    modelIn.close();
                } catch (IOException e) {
                }
            }
        }
        return model;
    }

    public String[] getSentencesFromFile(String inputFile) {
        String data = FileUtils.getFileDataAsString(inputFile);
        return sentenceDetector.sentDetect(data);
    }

    public String[] getSentences(String data) {
        return sentenceDetector.sentDetect(data);
    }

}
}
```

main class:

```
public class Main {
 public static void main(String args[]) {
  SentenceDetectorUtil util = new SentenceDetectorUtil(
    "path//to//your//en-sent.bin");

  String data = "Welcome to Stackoverflow Documentation.This is the first example in OenNLP.";

  String[] sentences = util.getSentences(data);

  for (String s : sentences)
   System.out.println(s +"\n");
 }
}
```

output will be:

> Welcome to Stackoverflow Documentation.
>
> This is the first example in OpenNLP.

Read OpenNLP online: https://riptutorial.com/nlp/topic/6052/opennlp

# Chapter 4: Sentence boundary detection in Python

## Examples

### With Stanford CoreNLP, from Python

You first need to run a Stanford CoreNLP server:

```
java -mx4g -cp "*" edu.stanford.nlp.pipeline.StanfordCoreNLPServer -port 9000 -timeout 50000
```

Here is a code snippet showing how to pass data to the Stanford CoreNLP server, using the `pycorenlp` Python package.

```
from pycorenlp import StanfordCoreNLP
import pprint

if __name__ == '__main__':
    nlp = StanfordCoreNLP('http://localhost:9000')
    fp = open("long_text.txt")
    text = fp.read()
    output = nlp.annotate(text, properties={
        'annotators': 'tokenize,ssplit,pos,depparse,parse',
        'outputFormat': 'json'
    })
    pp = pprint.PrettyPrinter(indent=4)
    pp.pprint(output)
```

### With python-ucto

Ucto is a rule-based tokeniser for multiple languages. It does sentence boundary detection as well. Although it is written in C++, there is a Python binding python-ucto to interface with it.

```
import ucto

#Set a file to use as tokeniser rules, this one is for English, other languages are available
too:
settingsfile = "/usr/local/etc/ucto/tokconfig-en"

#Initialise the tokeniser, options are passed as keyword arguments, defaults:
#   lowercase=False,uppercase=False,sentenceperlineinput=False,
#   sentenceperlineoutput=False,
#   sentencedetection=True, paragraphdetection=True, quotedetection=False,
#   debug=False
tokenizer = ucto.Tokenizer(settingsfile)

tokenizer.process("This is a sentence. This is another sentence. More sentences are better!")

for sentence in tokenizer.sentences():
    print(sentence)
```

## Using NLTK Library

You can find more info about Python Natural Language Toolkit (NLTK) sentence level tokenizer on their wiki.

From your command line:

```
$ python
>>> import nltk
>>> sent_tokenizer = nltk.tokenize.PunktSentenceTokenizer()
>>> text = "This is a sentence. This is another sentence. More sentences are better!"
>>> sent_tokenizer.tokenize(text)
Out[4]:
['This is a sentence.',
 'This is another sentence.',
 'More sentences are better!']
```

Read Sentence boundary detection in Python online:
https://riptutorial.com/nlp/topic/3833/sentence-boundary-detection-in-python

# Credits

| S. No | Chapters | Contributors |
|---|---|---|
| 1 | Getting started with nlp | Community, Gabor Angeli |
| 2 | N-GRAMS | M Monis Ahmed Khan, thepurpleowl |
| 3 | OpenNLP | caffeinator13 |
| 4 | Sentence boundary detection in Python | cgl, Franck Dernoncourt, JGreenwell, proycon |