# Part 11: Step by Step Guide to Master NLP – Syntactic Analysis

CHIRAG GOYAL — June 24, 2021

Advanced NLP

This article was published as a part of the Data Science Blogathon

## Introduction

This article is part of an ongoing blog series on Natural Language Processing (NLP). In the previous article, we discussed an entity extraction technique named i.e, Named Entity Recognition. There is also another entity extraction technique which is also a popular technique named **Topic Modeling**, which we will discuss in the subsequent articles of our blog series.

So, In this article, we will deep dive into Syntactic Analysis, which is one of the crucial levels of NLP.

**This is part-11 of the blog series on the Step by Step Guide to Natural Language Processing.**

## Table of Contents

## What is Syntactic analysis?

Syntactic analysis is defined as analysis that tells us the logical meaning of certainly given sentences or parts of those sentences. We also need to consider rules of grammar in order to define the logical meaning as well as the correctness of the sentences.

Or, In simple words, Syntactic analysis is the process of analyzing natural language with the rules of formal grammar. We applied grammatical rules only to categories and groups of words, not applies to individual words.

The syntactic analysis basically assigns a semantic structure to text. It is also known as syntax analysis or parsing. The word 'parsing' is originated from the Latin word 'pars' which means 'part'. The syntactic analysis deals with the syntax of Natural Language. In syntactic analysis, grammar rules have been used.

```
Sentence: School go a boy
```

The above sentence does not logically convey its meaning, and its grammatical structure is not correct. So, Syntactic analysis tells us whether a particular sentence conveys its logical meaning or not and whether its grammatical structure is correct or not.

As we discussed the steps or different levels of NLP, the third level of NLP is Syntactic analysis or parsing or syntax analysis. The main aim of this level is to draw exact meaning, or in simple words, you can say finding a dictionary meaning from the text. Syntax analysis checks the text for meaningfulness compared to the rules of formal grammar.

**For Example,** consider the following sentence

```
Sentence: “hot ice cream”
```

The above sentence would be rejected by the semantic analyzer.

Now, let’s define the Syntactic analysis formally,

In the above sense, syntactic analysis or parsing may be defined as the process of analyzing the strings of symbols in natural language conforming to the rules of formal grammar.

# Difference between Lexical and Syntactic analysis

The aim of lexical analysis is in Data Cleaning and Feature Extraction with the help of techniques such as

- Stemming,
- Lemmatization,
- Correcting misspelled words, etc.

But on the contrary, in syntactic analysis, our target is to :

- Find the roles played by words in a sentence,
- Interpret the relationship between words,
- Interpret the grammatical structure of sentences.

Let’s consider the following example having 2 sentences:

```
Sentences:
Patna is the capital of Bihar.
Is Patna the of Bihar capital?
```

In both sentences, all the words are the same, but only the first sentence is syntactically correct and easily understandable.

But we cannot make these distinctions using Basic lexical processing techniques. Therefore, we require more sophisticated syntax processing techniques to understand the relationship between individual words in a sentence.

The syntactical analysis looks at the following aspects in the sentence which lexical doesn’t :

## Words Order and Meaning

The syntactical analysis aims to extract the dependency of words with other words in the document. If we change the order of the words, then it will make it difficult to comprehend the sentence.

## Retaining Stop-Words

If we remove the stop-words, then it can altogether change the meaning of a sentence.

## Parts-of-speech of Words in a Sentence

Identifying the correct part-of-speech of a word is important.

**For Example,** Consider the following phrases:

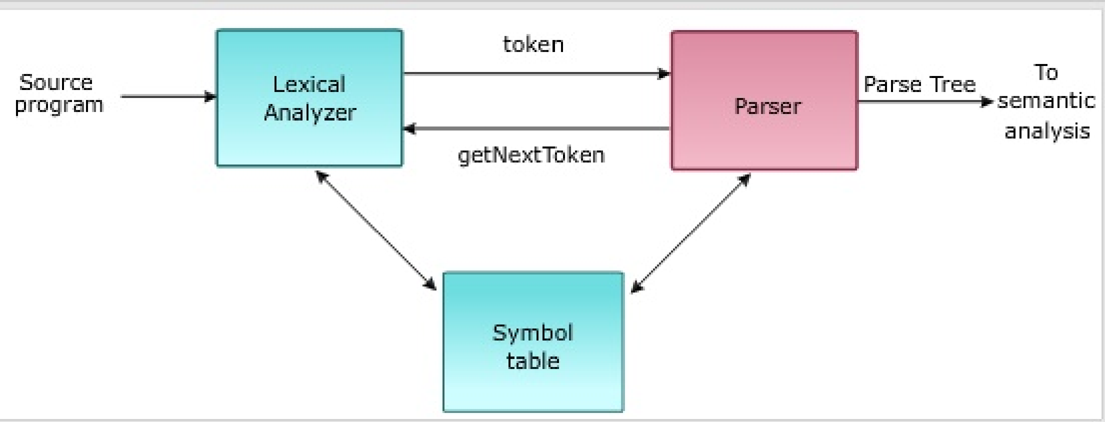
```
'cuts on his hand' (Here 'cuts' is a noun)
'he cuts an pineapple' (Here, 'cuts' is a verb)
```

# What is a Parser?

The parser is used to implement the task of parsing.

**Now, let's see what exactly is a Parser?**

It is defined as the software component that is designed for taking input text data and gives a structural representation of the input after verifying for correct syntax with the help of formal grammar. It also generates a data structure generally in the form of a parse tree or abstract syntax tree or other hierarchical structure.



**Interaction Between Lexical Analyzer and Parser**

**Image Source: Google Images**

We can understand the relevance of parsing in NLP with the help of the following points:

- The parser can be used to report any syntax error.
- It helps to recover from commonly occurring errors so that the processing of the remainder of the program can be continued.
- A parse tree is created with the help of a parser.
- The parser is used to create a symbol table, which plays an important role in NLP.
- A parser is also used to produce intermediate representations (IR).

# Different types of Parsers

As discussed, Basically, a parser is a procedural interpretation of grammar. It tries to find an optimal tree for a particular sentence after searching through the space of a variety of trees.

Let's discuss some of the available parsers:

- Recursive Descent Parser
- Shift-reduce Parser
- Chart Parser
- Regexp Parser

## Recursive Descent Parser

It is one of the most straightforward forms of parsing. Some important points about recursive descent parser are as follows:

- It follows a top-down process.
- It tries to check whether the syntax of the input stream is correct or not.

- The necessary operation for these types of parsers is to scan characters from the input stream and match them with the terminals with the help of grammar.

### Shift-reduce Parser

Some of the important points about shift-reduce parser are as follows:

- It follows a simple bottom-up process.
- It aims to find the words and phrases sequence that corresponds to the right-hand side of a grammar production and replaces them with the left-hand side of the production.
- It tries to find a word sequence that continues until the whole sentence is reduced.
- In simple words, this parser starts with the input symbol and aims to constructs the parser tree up to the start symbol.

### Chart Parser

Some of the important points about chart parser are as follows:

- Mainly, this parser is useful for ambiguous grammars, including grammars of natural languages.
- It applies the concept of dynamic programming to the parsing problems.
- Because of dynamic programming, it stores partial hypothesized results in a structure called a 'chart'.
- The 'chart' can also be reused in different scenarios.

### Regexp Parser

It is one of the most commonly used parsers. Some of the important points about the Regexp parser are as follows:

- It uses a regular expression that is defined in the form of grammar on top of a POS-tagged string.
- Basically, it uses these regular expressions to parse the input sentences and produce a parse tree out of this.

## What is Derivation?

We need a sequence of production rules in order to get the input string. The derivation is a set of production rules. During parsing, we have to decide the non-terminal, which is to be replaced along with deciding the production rule with the help of which the non-terminal will be replaced.

### Types of Derivation

In this section, we will discuss the two types of derivations, which can be used to decide which non-terminal to be replaced with the production rule:

### Left-most Derivation

In the left-most derivation, the sentential form of input is scanned and replaced from the left to the right. In this case, the sentential form is known as the left-sentential form.

### Right-most Derivation

In the left-most derivation, the sentential form of input is scanned and replaced from right to left. In this case, the sentential form is called the right-sentential form.

## Types of Parsing

Derivation divides parsing into the followings two types :

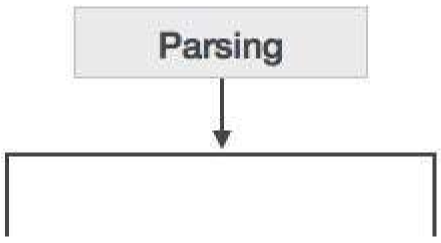


Image Source: Google Images

## Top-down Parsing

In top-down parsing, the parser starts producing the parse tree from the start symbol and then tries to transform the start symbol to the input. The most common form of top-down parsing uses the recursive procedure to process the input but its main disadvantage is backtracking .

## Bottom-up Parsing

In bottom-up parsing, the parser starts working with the input symbol and tries to construct the parser tree up to the start symbol.

# What is a Parse Tree?

It represents the graphical depiction of a derivation. The start symbol of derivation is considered the root node of the parse tree and the leaf nodes are terminals, and interior nodes are non-terminals.

The most useful property of the parse tree is that the in-order traversal of the tree will produce the original input string.

**For Example,** Consider the following sentence:

```
Sentence: the dog saw a man in the park
```

After analyzing the sentence, the parse tree formed is shown below:

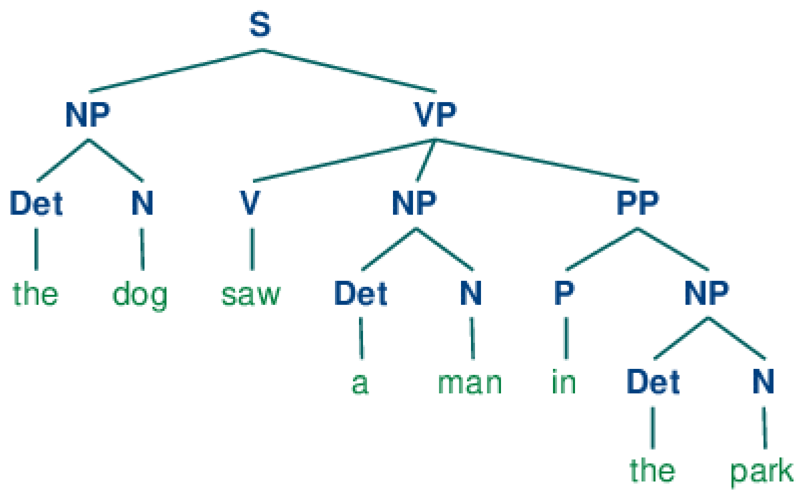


Image Source: Google Images

## This ends our Part-11 of the Blog Series on Natural Language Processing!

# Other Blog Posts by Me

You can also check my previous blog posts.

Previous Data Science Blog posts.

# LinkedIn

Here is my Linkedin profile in case you want to connect with me. I'll be happy to be connected with you.

# Email

For any queries, you can mail me on Gmail .

# End Notes