

[Home](#)

Part 12: Step by Step Guide to Master NLP – Grammar in NLP



[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 some basic concepts related to syntactic analysis. In that article, we covered concepts such as parsing, parse trees, and parsers, etc. But we not discussed the concept of grammar in that article. So, In continuation to that article, we will complete a Syntactic analysis in this article.

So, In this article, we will deep dive into the concept of Grammar with its types that are heavily used while we working on the level of Syntactic Analysis in NLP.

This is part-12 of the blog series on the Step by Step Guide to Natural Language Processing.

Table of Contents

1. What is Grammar?

2. Different Types of Grammar in NLP

- Context-Free Grammar (CFG)
- Constituency Grammar (CG)
- Dependency Grammar (DG)

What is Grammar?

Grammar is defined as the rules for forming well-structured sentences.

While describing the syntactic structure of well-formed programs, Grammar plays a very essential and important role. In simple words, Grammar denotes syntactical rules that are used for conversation in natural languages.

The theory of formal languages is not only applicable here but is also applicable in the fields of Computer Science mainly in programming languages and data structures.

For Example, in the ‘C’ programming language, the precise grammar rules state how functions are made with the help of lists and statements.

Mathematically, a grammar G can be written as a 4-tuple (N, T, S, P) where,

T or Σ = set of terminal symbols.

S = Start symbol where $S \in N$

P = Production rules for Terminals as well as Non-terminals.

It has the form $\alpha \rightarrow \beta$, where α and β are strings on $V_N \cup \Sigma$ and at least one symbol of α belongs to V_N

Context-Free Grammar (CFG)

A context-free grammar, which is in short represented as CFG, is a notation used for describing the languages and it is a superset of Regular grammar which you can see from the following diagram:

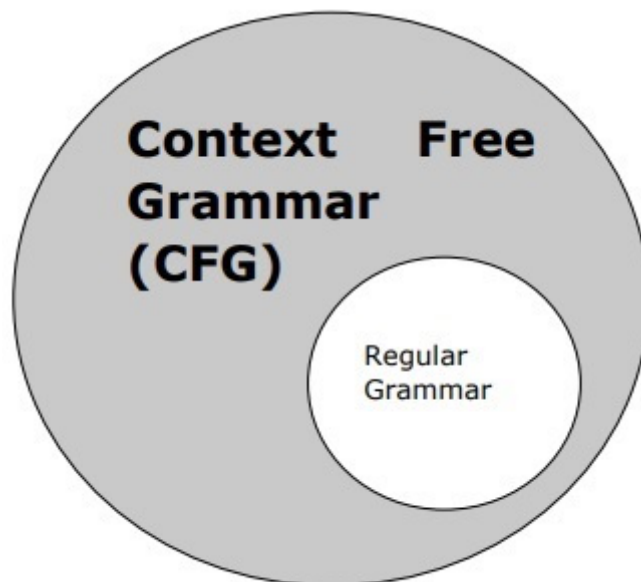


Image Source: Google Images

CFG consists of a finite set of grammar rules having the following four components

- Set of Non-Terminals
- Set of Terminals
- Set of Productions
- Start Symbol

Set of Non-terminals

It is represented by V . The non-terminals are syntactic variables that denote the sets of strings, which helps in defining the language that is generated with the help of grammar.

Set of Terminals

It is also known as tokens and represented by Σ . Strings are formed with the help of the basic symbols of terminals.

Set of Productions

It is represented by P . The set gives an idea about how the terminals and nonterminals can be combined. Every production consists of the following components:

- Non-terminals,
- Arrow,
- Terminals (the sequence of terminals).

The left side of production is called non-terminals while the right side of production is called terminals.

Start Symbol

Constituency Grammar (CG)

It is also known as Phrase structure grammar. It is called constituency Grammar as it is based on the constituency relation. It is the opposite of dependency grammar.

Before deep dive into the discussion of CG, let’s see some fundamental points about constituency grammar and constituency relation.

- All the related frameworks view the sentence structure in terms of constituency relation.
- To derive the constituency relation, we take the help of subject-predicate division of Latin as well as Greek grammar.
- Here we study the clause structure in terms of noun phrase NP and verb phrase VP.

For Example,

Sentence: This tree is illustrating the constituency relation

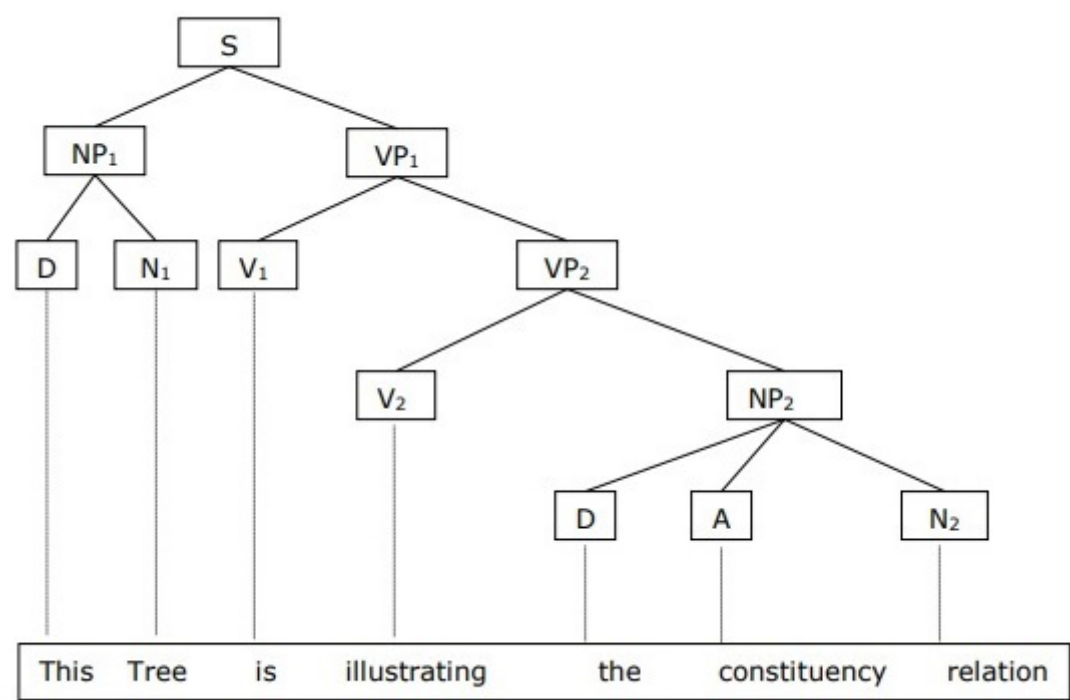


Image Source: Google Images

Now, Let’s deep dive into the discussion on Constituency Grammar:

In Constituency Grammar, the constituents can be any word, group of words, or phrases and the goal of constituency grammar is to organize any sentence into its constituents using their properties. To derive these properties we generally take the help of:

- Part of speech tagging,
- A noun or Verb phrase identification, etc

For Example, constituency grammar can organize any sentence into its three constituents- a subject, a context, and an object.

Sentence: <subject> <context> <object>

These three constituents can take different values and as a result, they can generate different sentences. For Example, If we have the following constituents, then

<subject> The horses / The dogs / They
<context> are running / are barking / are eating
<object> in the park / happily / since the morning

Example sentences that we can be generated with the help of the above constituents are:

“The dogs are barking in the park”
“They are eating happily”
“The horses are running since the morning”

Now, let’s look at another view of constituency grammar is to define their grammar in terms of their part of speech tags.

Say a grammar structure containing a

[determiner, noun] [adjective, verb] [preposition, determiner, noun]

which corresponds to the same sentence – “The dogs are barking in the park”

Another view (Using Part of Speech)
< DT NN > < JJ VB > < PRP DT NN > -----> The dogs are barking in the park

Dependency Grammar (DG)

It is opposite to the constituency grammar and is based on the dependency relation. Dependency grammar (DG) is opposite to constituency grammar because it lacks phrasal nodes.

Before deep dive into the discussion of DG, let’s see some fundamental points about Dependency grammar and Dependency relation.

- In Dependency Grammar, the words are connected to each other by directed links.
- The verb is considered the center of the clause structure.
- Every other syntactic unit is connected to the verb in terms of directed link. These syntactic units are called dependencies.

For Example,

Sentence: This tree is illustrating the dependency relation

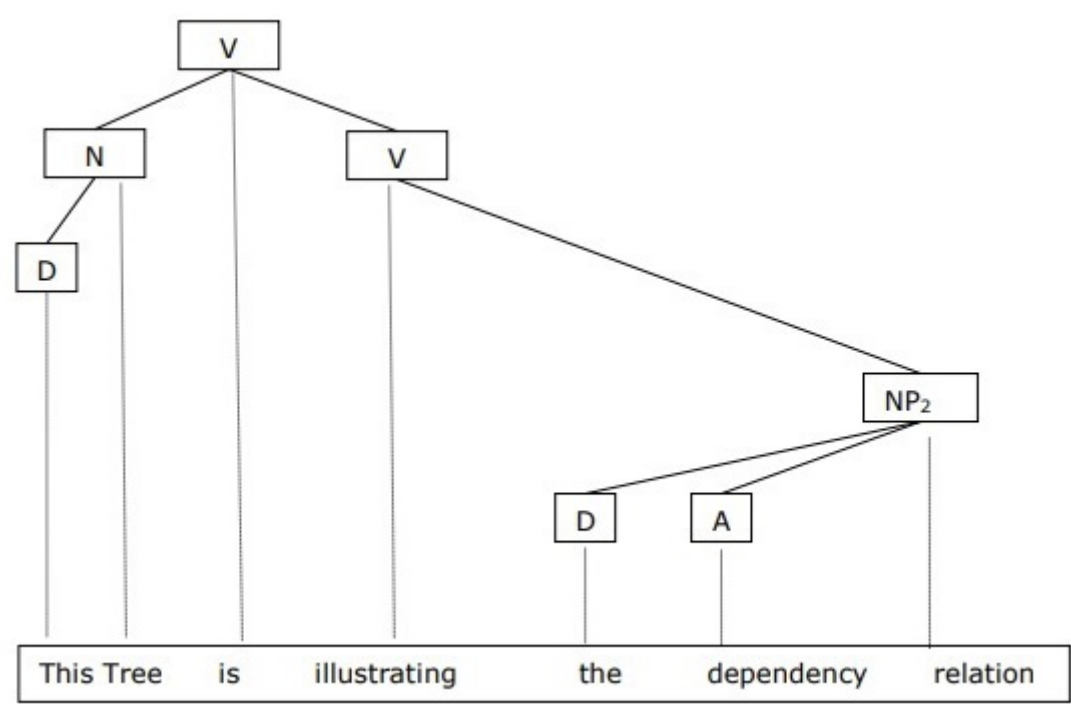


Image Source: Google Images

Now, Let’s deep dive into the discussion of Dependency Grammar:

1. Dependency Grammar states that words of a sentence are dependent upon other words of the sentence.

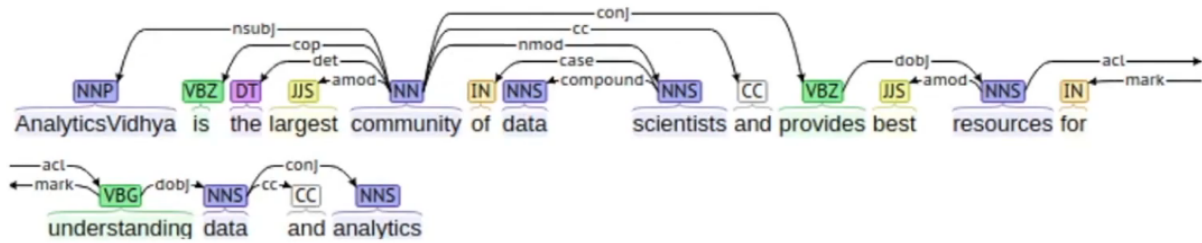
For Example, in the previous sentence which we discussed in CG, “barking dog” was mentioned and the dog was modified with the help of barking as the dependency adjective modifier exists between the two.

2. It organizes the words of a sentence according to their dependencies. One of the words in a sentence behaves as a root and all the other words except that word itself are linked directly or indirectly with the root using their dependencies. These dependencies represent relationships among the words in a sentence and dependency grammars are used to infer the structure and semantic dependencies between the words.

For Example, Consider the following sentence:

Sentence: Analytics Vidhya is the largest community of data scientists and provides the best resources for understanding data and analytics

The dependency tree of the above sentence is shown below:



In the above tree, the root word is “community” having NN as the part of speech tag and every other word of this tree is connected to root, directly or indirectly, with the help of dependency relation such as a direct object, direct subject, modifiers, etc.

These relationships define the roles and functions of each word in the sentence and how multiple words are connected together.

We can represent every dependency in the form of a triplet which contains a governor, a relation, and a dependent,

Relation : (Governor, Relation, Dependent)

which implies that a dependent is connected to the governor with the help of relation, or in other words, they are considered the subject, verb, and object respectively.

For Example, Consider the following same sentence again:

Sentence: Analytics Vidhya is the largest community of data scientists

Then, we separate the sentence in the following manner:

< Analyticsvidhya> <is> <the largest community of data scientists>

Now, let’s identify different components in the above sentence:

- **Subject:** “Analytics Vidhya” is the subject and is playing the role of a governor.
- **Verb:** “is” is the verb and is playing the role of the relation.
- **Object:** “the largest community of data scientists” is the dependent or the object.

Some use cases of Dependency grammars are as follows:

Named Entity Recognition

It can be used to solve the problems related to named entity recognition (NER).

Question Answering System

It can be used to understand the relational and structural aspects of question-answering systems.

Text summarization and Text classification

It can also be used for text summarization problems and they are also used as features for text classification problems.

NOTE:

The Parse tree that uses Constituency grammar is called the constituency-based parse tree, and the Parse tree that uses dependency grammar is called the dependency-based parse tree.

This ends our Part-12 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

Thanks for reading!

I hope that you have enjoyed the article. If you like it, share it with your friends also. Something not mentioned or want to share your thoughts? Feel free to comment below And I'll get back to you. 😊

The media shown in this article are not owned by Analytics Vidhya and are used at the Author's discretion.

[Part 11: Step by Step Guide to Master NLP - Syntactic Analysis](#)

[Part 2: Step by Step Guide to NLP - Knowledge Required to Learn NLP](#)

[How to Perform Basic Text Analysis without Training Dataset](#)

[blogathon](#) [Grammar in NLP](#)

About the Author



[CHIRAG GOYAL](#)

Our Top Authors

