**Project Subtopic:** Display Parse Tree using NLTK

**Implemented by:** Sagar Dalwadi (sagardd2)

**Program code file:** nltk\_parse\_tree.py

**Program Input data:** ParseTree\_InputData

**Program Output:** Screenshots named – nltk\_pt\_ss1 to nltk\_pt\_ss4

1) An overview of the function of the code (i.e., what it does and what it can be used for).

**Background context for a Project topic:**

What is Syntax?  
A natural language typically follows a hierarchical structure, and contains the following components: 

* Sentences
* Clauses
* Phrases
* Words

Syntax refers to the set of rules, principles, processes that govern the structure of sentences in a natural language. One basic description of syntax is how different words such as Subject, Verbs, Nouns, Noun Phrases, etc. are sequenced in a sentence.

Some of the syntactic categories of a natural language are as follows:

* Sentence(S)
* Noun Phrase(NP)
* Determiner(Det)
* Verb Phrase(VP)
* Prepositional Phrase(PP)
* Verb(V)
* Noun(N)

Syntax Tree:   
A Syntax tree or a parse tree is a tree representation of different syntactic categories of a sentence. It helps us to understand the syntactical structure of a sentence.

**Objective (What it does):**

* I have implemented a python program using NLTK library to display a parse tree for a given sentence in English that helps visualize the syntactic structure of a sentence.
* The program imports an NLTK toolkit into python and uses the pre-defined libraries to extract the syntactic categories from a sentence, print these categories and finally display a parse tree for the sentence.

**Purpose (What it can be used for):**

This can be used as:

* An independent python program in your application extending your application need to either display a parse tree for the input data it’s receiving from its upstream or extracting the syntactic categories of a sentence in your NLP specific application to further apply other NLP techniques.
* Can be converted into a python function that can be incorporated into python bindings of MeTA (metapy) and act as an extension to the MeTA library for NLP.

2) Documentation of how the software is implemented with sufficient detail so that others can have a basic understanding of your code for future extension or any further improvement.

* My program is implemented using an open source toolkit available to build python programs called Natural language Toolkit (NLTK). It provides easy to use interfaces and suite of text processing libraries for different NLP techniques.
* First import required libraries corresponding to the function of the application along with nltk library
* NLTK provides set of default datasets available that can be imported for the use in your application. These datasets can be downloaded using nltk.download function.
* In order to use functions within NLTK, import the required libraries
* At this step an input is required for your program. I have declared a variable to provide a sentence for which parse tree is to be displayed. You can also integrate directly with a front-end application that provides an input data or read the input data from a file as well.
* Now use the functions available from NLTK library to extract the different syntactic categories from a sentence and then print these categories as well as display a parse tree.
* As a further enhancement to this program these categories can be provided as an input to your software for the next steps of NLP.

3) Documentation of the usage of the software including either documentation of usages of APIs or detailed instructions on how to install and run a software, whichever is applicable.

* First of all, ensure you have python as well as NLTK installed on your system. Below are the instructions on how to install NLTK:

NLTK requires Python versions 3.6, 3.7, 3.8, or 3.9

**Mac/Unix :**

1. Install NLTK: run sudo pip install -U nltk
2. Install Numpy (optional): run sudo pip install -U numpy
3. Test installation: run python then type import nltk

**Windows :**

1. Install Numpy (optional): <http://sourceforge.net/projects/numpy/files/NumPy/> (the version that specifies pythnon3.5)
2. Install NLTK: <http://pypi.python.org/pypi/nltk>
3. Test installation: Start>Python35 , then type import nltk

* You can setup python virtual environments if you want to run this program
* Below are the steps in order to run the program:
  1. First, provide an input sentence for which you would like to display a parse tree. Open a program in python editor and update/provide your sentence in the variable text\_sentence.
  2. Go to terminal and execute the command python nltk\_parse\_tree.py
  3. Once the program is executed successfully it would open a new nltk window and display the parse tree for the sentence provided as an input.

4) Brief description of contribution of each team member in case of a multi-person team.

Our project proposal included 2 different subtopics for system extension to implement keyword extraction using SpaCy as well as display parse tree using NLTK.

* To distribute the work among the team members, we have divided based on the subtopics of the project.
* I have contributed towards the subtopic of displaying a parse tree using NLTK and implemented an end to end python program corresponding to this subtopic.