In my code I tried to keep the functions as simple as possible while using the tools offered by the spacy library. I also added a couple of “quality of life” functions in order to have a cleaner code. I also wrote comments for each function to explain briefly the purpose of the single function, but as follows I will do a general overview of the most important methods:

1. **dep\_paths(*sentence*)**: Takes in input a single sentence (*string*) and for each token of the parsed sentence (*Doc object*), use a recursive tree visit (*extract\_path()*) to retrieve the path starting from the root and ending in the selected token.
2. **ex\_subtrees(*sentence*)**: Takes in input a single sentence and for each token in the parsed sentence search for the subtree of that token using *extract\_subtree(token)*. Returns a list object containing all the subtrees (the subtrees are converted to a List object).
3. **subtree\_checking(*tokens, sentence*)**: This function takes in input a list of words (like [“I”, “am”, “Giorgio”]) and a sentence (String). First it extracts all the possible subtrees in the parsed sentence using *ex\_subtrees()*; then use *equality(subtree, tokens)* on every subtree in order to check if the words form one of the possible subtree in the sentence. Return a Boolean value.
4. **identify\_head(*span*)**: Given in input a string of words, simply returns the root of the parsed sentence.
5. **extract\_so(sentence)**: Given in input a sentence, creates a dictionary of lists which keys are *“nsubj”*, “*dobj*” and “*iobj*”; then fills the dictionary with the tokens corresponding to that type of dependency. Return the dictionary.
6. **assignment\_test()**: A test function to try all the features of the file. In the function is defined a sample sentence “I saw a man with a Telescope” that is used to try all the functions. This function prints all the results of the other relevant functions.