NLU FIRST ASSIGNMENT

Pietro Ghislieri, 223784

1. First part

- Importing SpaCy
- Inizialize the sentence and the en_core_web_sm SpaCy model
- I used doc=nlp(sentense) inside every function so that is possible to insert a sentence that is not been initialize in the beginning

```
In [1]: import spacy
        nlp = spacy.load("en_core_web_sm")
        sentence='I saw the man with a telescope'
```

2. Function firtspoint(sentence)

The first funcion as required extract a path of dependency relations from the ROOT to a token.

while not t.dep_=='ROOT':

    token_path.insert(0,t.dep_)

I used the method insert(position, object) to be able to insert the object always at the top of the list so that I didn't have to invert it.

I continue to insert path until I did find the ROOT, then I add the string 'ROOT' at the top of the list, save it as value of a dictionary that has as keys the values of the parsed sentence. Then I initialize the token_path list to an empty one to store the path for the next token.

3. Function secondpoint(sentence)

The second function extract subtree of a dependents given a token.

I iterate for all the token in the 'doc' variable, for each token I than used the function token.subtree to retrieve the subtree for each token and stored the result in a dictionary with the value as token.text not to have a <SpaCy.Token > object just to simplify the search in the next functions. As before the dictionary has as keys the values of the parsed sentence.

4.Function thirdpoint(sentence,sub)

The third function check if a given list of tokens (segment of a sentence) forms a subtree.

Using the function “secondpoint(sentence)” I obtain a dictionary with the subtrees for the tokens of the sentence than I check if the subsentence passed as a parameter is present in the dictionary.

5.Function fourthpoint(sentence, start=0, end=len(sentence))

This function identifies head of a span, given its tokens.

I use doc[:] to create a pan than find the span Head using span.root.

If no start and end of the sentence is given, I create the span of all the length of the sentence, but I also leave the possibility to give a start and end point.

6.Function Fifthpoint(sentence)

The function extract sentence subject, direct object and indirect object spans

I check the depth of every token if is equal to: 'nsubj' or 'dobj' or 'dative’ (‘iobj’) and if that’s the case I store the token in a dictionary that has as keys the sentence subject, direct object and indirect object spans.