In [2]:

```
import nltk

nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\xxx3k\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\xxx3k\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

Out[2]:

```
True
```

In [3]:

```
def get_pos_tags(sentence):
    tokens = nltk.word_tokenize(sentence)
    pos_tags = nltk.pos_tag(tokens)
    return pos_tags
```

In [4]:

```
sentence = "The quick brown fox jumps over the lazy dog."
pos_tags = get_pos_tags(sentence)
print(pos_tags)
```

```
[('The', 'DT'), ('quick', 'JJ'), ('brown', 'NN'), ('fox', 'NN'), ('jumps', 'V
BZ'), ('over', 'IN'), ('the', 'DT'), ('lazy', 'JJ'), ('dog', 'NN'), ('.',
'.')]
```

In [5]:

```
def get_sentence_structure(sentence):
    pos_tags = get_pos_tags(sentence)
    structure = {}
    for i in range(len(pos_tags)):
        if pos_tags[i][1].startswith('V'):
            structure['verb'] = pos_tags[i][0]
            if i > 0:
                structure['subject'] = pos_tags[i-1][0]
            if i < len(pos_tags)-1:
                structure['object'] = pos_tags[i+1][0]
    return structure
```

In [6]:

```
sentence = "The quick brown fox jumps over the lazy dog."
structure = get_sentence_structure(sentence)
print(structure)
```

```
{'verb': 'jumps', 'subject': 'fox', 'object': 'over'}
```

```
In [ ]:
```