

How to use WordNet in Python

 educative.io/answers/how-to-use-wordnet-in-python

WordNet is an English dictionary which is a part of **Natural Language Tool Kit** (NLTK) for Python. This is an extensive library built to make **Natural Language Processing** (NLP) easy. Some basic functions will be discussed in this article. To start using WordNet, you have to import it first:

```
from nltk.corpus import wordnet
```

Synsets and Lemmas

In WordNet, similar words are grouped into a set known as a **Synset** (short for **Synonym-set**). Every Synset has a name, a part-of-speech, and a number. The words in a Synset are known as **Lemmas**.

Getting Synsets

The function `wordnet.synsets('word')` returns an array containing all the Synsets related to the word passed to it as the argument.

Example:

```
print(wordnet.synsets('room'))
```

Output:

```
[Synset('room.n.01'), Synset('room.n.02'), Synset('room.n.03'), Synset('room.n.04'),  
Synset('board.v.02')]
```

The method returned five Synsets; four have the name **'room'** and are a **nouns**, while the last one's name is **'board'** and is a **verb**. The output also suggests that the word **'room'** has a total of five meanings or contexts.

Getting definition of a Synset

By using `definition()`, a single Synset can further be explored for a definition common to all the **Lemmas** it contains. This method returns a string, which is the **common** definition. There are two ways to do this:

1. We can use the array returned by `synsets('word')` and access one of its elements:

```
syn_arr = wordnet.synsets('room')  
syn_arr[0].definition()
```

Output :

an area within a building enclosed by walls and floor and ceiling

2. Or, pass the name of the Synset, its part-of-speech and its number, to `synset()` and then use `definition()`:

```
wordnet.synset('room.n.02').definition()
```

Output :

space for movement

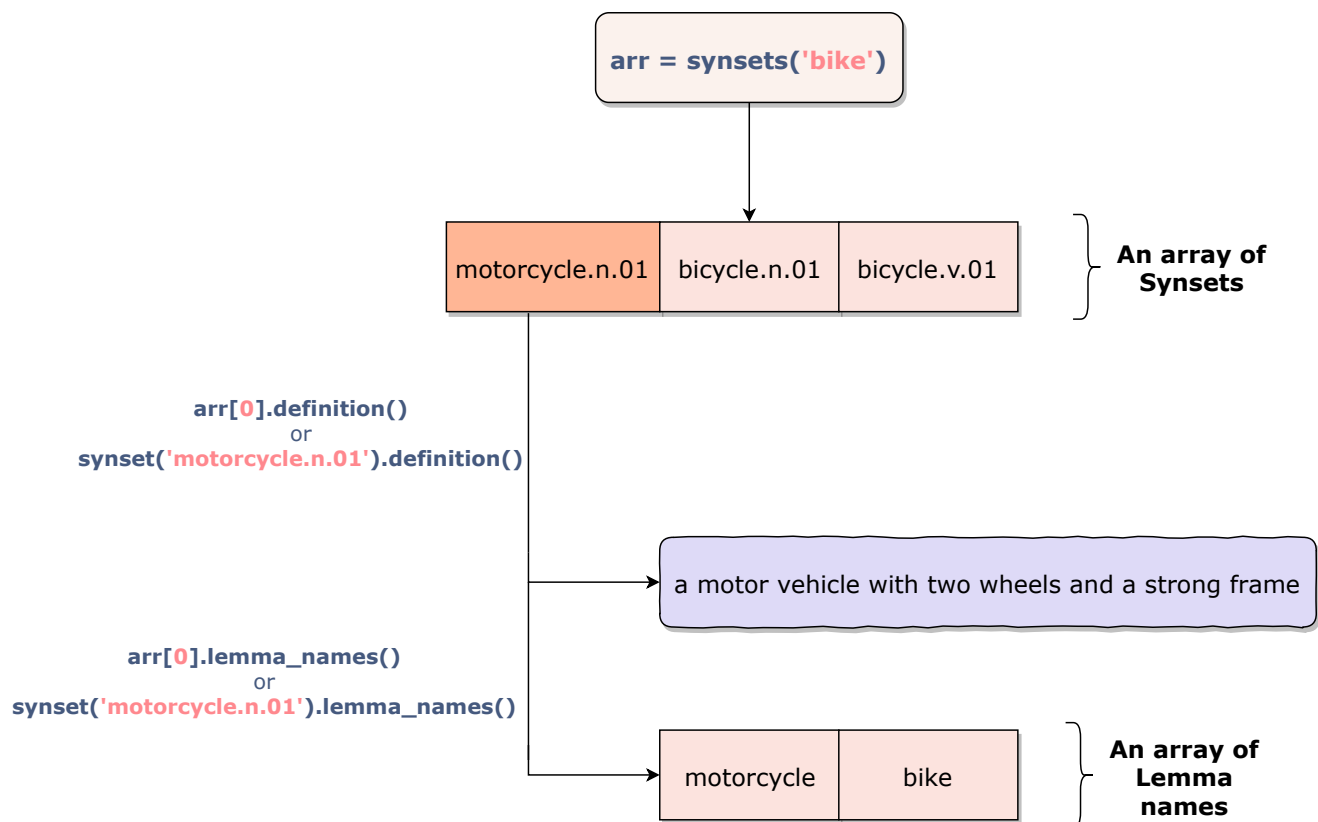
Getting all Lemmas of a Synset

Similarly, `lemma_names()` can be used in two ways to **return an array** of all Lemma names:

```
print(syn_arr[1].lemma_names())  
# or  
print(wordnet.synset('board.v.02').lemma_names())
```

Output :

```
[u'room', u'way', u'elbow_room']  
[u'board', u'room']
```



Using `synsets()`, `synset()`, `definition()` and `lemma_names()`

Hyponyms

A **Hyponym** is a specialisation of a Synset. It can be thought of as a child (or derived) class in inheritance. The function `hyponyms()` returns an array containing all the Synsets which are Hyponyms of the given Synset:

```
print(wordnet.synset('calendar.n.01').hyponyms())
```

Output :

```
[Synset('lunar_calendar.n.01'), Synset('lunisolar_calendar.n.01'),  
Synset('solar_calendar.n.01')]
```

Hypernyms

A **Hypernym** is a generalisation of a Synset (i.e. the opposite of a Hyponym). An array containing all Hypernyms of a Synset is returned by `hypernyms()`:

```
print(wordnet.synset('solar_calendar.n.01').hypernyms())
```

Output :

```
[Synset('calendar.n.01')]
```

Copyright ©2023 Educative, Inc. All rights reserved