

```
# Word net stores definitions of words
# Lemmas which are synonyms
# examples which give the usage of the word
# hypernyms which is a higher or less specific word similar
# hyponyms which is lower or more specific of a word
```

```
import nltk
nltk.download('wordnet')
nltk.download('omw-1.4')
nltk.download('sentiwordnet')
nltk.download('gutenberg')
nltk.download('genesis')
nltk.download('inaugural')
nltk.download('nps_chat')
nltk.download('treebank')
nltk.download('webtext')
nltk.download('stopwords')
```

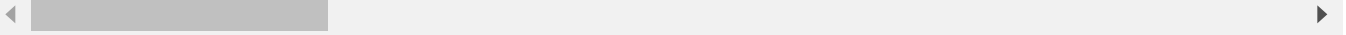
```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data] Package omw-1.4 is already up-to-date!
[nltk_data] Downloading package sentiwordnet to /root/nltk_data...
[nltk_data] Package sentiwordnet is already up-to-date!
[nltk_data] Downloading package gutenberg to /root/nltk_data...
[nltk_data] Package gutenberg is already up-to-date!
[nltk_data] Downloading package genesis to /root/nltk_data...
[nltk_data] Package genesis is already up-to-date!
[nltk_data] Downloading package inaugural to /root/nltk_data...
[nltk_data] Package inaugural is already up-to-date!
[nltk_data] Downloading package nps_chat to /root/nltk_data...
[nltk_data] Package nps_chat is already up-to-date!
[nltk_data] Downloading package treebank to /root/nltk_data...
[nltk_data] Package treebank is already up-to-date!
[nltk_data] Downloading package webtext to /root/nltk_data...
[nltk_data] Package webtext is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
True
```

```
from nltk.corpus import wordnet as wn
noun = "rat"
wn.synsets(noun,pos=wn.NOUN)
```

```
[Synset('rat.n.01'),
 Synset('scab.n.01'),
 Synset('rotter.n.01'),
 Synset('informer.n.01'),
 Synset('rat.n.05')]
```

```
noun_synset= wn.synsets(noun,pos=wn.NOUN)[2]
print(noun_synset.lemmas())
print(noun_synset.definition())
print(noun_synset.examples())

[Lemma('rotter.n.01.rotter'), Lemma('rotter.n.01.dirty_dog'), Lemma('rotter.n.01.rat'),
a person who is deemed to be despicable or contemptible
['only a rotter would do that', 'kill the rat', 'throw the bum out', 'you cowardly littl
```



```
# Hypernyms for nouns all go back to entity
# So there is a top level for hypernyms for all nouns
nouns = wn.synsets(noun,pos=wn.NOUN)[0]
while nouns.hypernyms():
    nouns = nouns.hypernyms()[0]
print(nouns)
```

```
Synset('rodent.n.01')
Synset('placental.n.01')
Synset('mammal.n.01')
Synset('vertebrate.n.01')
Synset('chordate.n.01')
Synset('animal.n.01')
Synset('organism.n.01')
Synset('living_thing.n.01')
Synset('whole.n.02')
Synset('object.n.01')
Synset('physical_entity.n.01')
Synset('entity.n.01')
```

```
print(nouns.hyponyms())
print(nouns.hypernyms())
print(nouns.part_meronyms())
print(nouns.part_holonyms())
print(nouns.lemmas()[0].antonyms())
```

```
[Synset('abstraction.n.06'), Synset('physical_entity.n.01'), Synset('thing.n.08')]
[]
[]
[]
[]
```

```
verb = "run"
wn.synsets(verb,pos=wn.VERB)
```

```
[Synset('run.v.01'),
Synset('scat.v.01'),
Synset('run.v.03'),
Synset('operate.v.01'),
Synset('run.v.05'),
```

```

Synset('run.v.06'),
Synset('function.v.01'),
Synset('range.v.01'),
Synset('campaign.v.01'),
Synset('play.v.18'),
Synset('run.v.11'),
Synset('tend.v.01'),
Synset('run.v.13'),
Synset('run.v.14'),
Synset('run.v.15'),
Synset('run.v.16'),
Synset('prevail.v.03'),
Synset('run.v.18'),
Synset('run.v.19'),
Synset('carry.v.15'),
Synset('run.v.21'),
Synset('guide.v.05'),
Synset('run.v.23'),
Synset('run.v.24'),
Synset('run.v.25'),
Synset('run.v.26'),
Synset('run.v.27'),
Synset('run.v.28'),
Synset('run.v.29'),
Synset('run.v.30'),
Synset('run.v.31'),
Synset('run.v.32'),
Synset('run.v.33'),
Synset('run.v.34'),
Synset('ply.v.03'),
Synset('hunt.v.01'),
Synset('race.v.02'),
Synset('move.v.13'),
Synset('melt.v.01'),
Synset('ladder.v.01'),
Synset('run.v.41')]

```

```

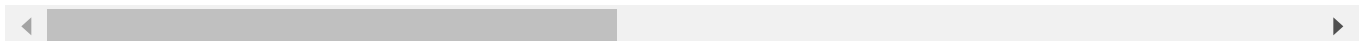
verb_synset= wn.synsets(verb,pos=wn.VERB)[2]
print(verb_synset.lemmas())
print(verb_synset.definition())
print(verb_synset.examples())

```

```

[Lemma('run.v.03.run'), Lemma('run.v.03.go'), Lemma('run.v.03.pass'), Lemma('run.v.03.1e
stretch out over a distance, space, time, or scope; run or extend between two points or
['Service runs all the way to Cranbury', "His knowledge doesn't go very far", 'My memory

```



```

# This hypernym set ends earlier
# All verbs aren't as related as nouns
# no top level for verbs
verbs = wn.synsets(verb,pos=wn.VERB)[0]
while verbs.hypernyms():
    verbs = verbs.hypernyms()[0]

```

```
print(verbs)

Synset('travel_rapidly.v.01')
Synset('travel.v.01')

wn.morphy(verb,wn.VERB)

'run'

lion = wn.synset('lion.n.01')
cat = wn.synset('cat.n.01')
print(lion.path_similarity(cat))

0.25

# it seems like there are differnt ways to see
# if two words are similar using Wu Palmer is said
# that they are more similar than path similarity
wn.wup_similarity(lion,cat)

0.896551724137931

from nltk.wsd import lesk
print(lesk(lion.definition(),'cat','n'))

Synset('kat.n.01')

from nltk.corpus.reader import sentiwordnet
# SentiwordNet is used for sentimental analysis
# So this would be if a text is either Positive or Negative
# As well as how objective is the text
# this could be used to see how positive or negative some perception is
# like if a politician does something we can see from peoples reactions on
# social media if it was good or bad
from nltk.corpus import sentiwordnet as swn
sent = "The cow jumped over the hill"
tokens = sent.split()
neg = 0
pos = 0
obj = 0
for token in tokens:
    syn_list = list(swn.senti_synsets(token))
    if syn_list:
        syn = syn_list[0]
        neg += syn.neg_score()
        pos += syn.pos_score()
```

```
obj += syn.obj_score()
print("Negative score = " + str(neg))
print("Positive score = " + str(pos))
print("Objective score = " + str(obj))
```

```
Negative score = 0.0
Positive score = 0.0
Objective score = 4.0
```

```
# Collocation is when two or more words combine
# you won't be able to switch out the word for
# another similar word to get the same meaning
from nltk.book import text4
text4.collocations()
```

```
United States; fellow citizens; years ago; four years; Federal
Government; General Government; American people; Vice President; God
bless; Chief Justice; one another; fellow Americans; Old World;
Almighty God; Fellow citizens; Chief Magistrate; every citizen; Indian
tribes; public debt; foreign nations
```

[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 10:40 PM

● ✕