```python
 1  #%% md
 2  Summary of WordNet:
 3
 4  WordNet is a hierarchical organization of nouns,
    verbs, adjectives, and adverbs. It contains short
    definitions of words called Glosses. It also contains
     Synsets, which are sets of synonyms for a word.
    Lastly, it also provides examples for using a word.
 5  #%% md
 6  First need to import wordNet:
 7  #%% md
 8
 9  #%%
10  from nltk.corpus import wordnet as wn
11  from nltk.wsd import lesk
12  from nltk.corpus import sentiwordnet as swn
13  #%% md
14  Select a noun and print out its synsets:
15  #%%
16  wn.synsets("book")
17  #%% md
18  Select a synset, and get its definition, usage
    examples, and lemmas:
19  #%%
20  #definition
21  wn.synset('book.n.01').definition()
22  #%%
23  #examples
24  wn.synset('book.n.01').examples()
25  #%%
26  #lemmas
27  wn.synset('book.n.01').lemmas()
28  #%% md
29  For the selected synset traverse up the WordNet
    hierarchy:
30  #%%
31  book = wn.synset('book.n.01')
32  hyp = book.hypernyms()[0]
33  top = wn.synset('entity.n.01')
34  while hyp:
35      print(hyp)
```

```
36        if hyp == top:
37            break
38        if hyp.hypernyms():
39            hyp = hyp.hypernyms()[0]
40 #%% md
41 WordNet Organization for Nouns:
42 For each noun WordNet has the hypernym and gradually
   goes all the way up to the most abstract term which
   is entity. Book went to publication, then work, then
   product and so on getting more and more abstract as
   it goes up the hierarchy. Entity being very generic
   and at the top.
43 #%% md
44 Output the list of hypernyms, hyponyms, meronyms,
   holonyms, and antonyms for the noun:
45 #%%
46 #hypernyms
47 book = wn.synset('book.n.01')
48 print("hypernyms: ", book.hypernyms())
49 #%%
50 #hyponyms
51 book = wn.synset('book.n.01')
52 print("hyponyms: ", book.hyponyms())
53 #%%
54 #meronyms
55 book = wn.synset('book.n.01')
56 print("meronyms: ", book.member_meronyms())
57 #%%
58 #holonyms
59 book = wn.synset('book.n.01')
60 print("holoynms: ", book.part_holonyms())
61 #%%
62 #antonyms
63 book = wn.synset('book.n.01')
64 print("antonyms: ", book.lemmas()[0].antonyms())
65 #%% md
66 Select a verb and output all synsets:
67 #%%
68 wn.synsets("ran")
69 #%% md
70 Select a synset, and get its definition, usage
```

```
70 examples, and lemmas:
71 #%%
72 #definition
73 wn.synset('run.v.01').definition()
74 #%%
75 #examples
76 wn.synset('run.v.01').examples()
77 #%%
78 #lemmas
79 wn.synset('run.v.01').lemmas()
80 #%% md
81 For the selected synset traverse up the WordNet
   hierarchy:
82 #%%
83 run = wn.synset('run.v.01')
84 hyp = run.hypernyms()[0]
85 top = wn.synset('entity.n.01')
86 while hyp:
87     print(hyp)
88     if hyp == top:
89         break
90     if hyp.hypernyms():
91         hyp = hyp.hypernyms()[0]
92 #%% md
93 WordNet Organization for Verbs:
94 Unlike for nouns, there is no uniform top level
   synset. So, trying to traverse up the WordNet
   hierarchy doesn't really work.
95 #%%
96 #use morphy
97 wn.morphy('run', wn.VERB)
98 #%%
99 #select two similar words:
100 orange = wn.synset('orange.n.01')
101 strawberry = wn.synset('strawberry.n.01')
102
103 #Run Wu-Palmer
104 wn.wup_similarity(orange, strawberry)
105 #%%
106 #run lesk algorithm
107 sent = ['I', 'won', 'money', 'playing', 'poker', '.'
```

```
107 ]
108 print(lesk(sent, 'poker', 'n'))
109 #%% md
110 The Wu-Palmer metric shows that orange and
    strawberry have a lot of common ancestor words. A
    score of 0.75 is pretty high.
111 #%% md
112 SentiWordNet:
113 SentiWordNet is a tool built on top of WordNet that
    does the same things wordNet can do, but in addition
     also has 3 scores for each synset: positivity,
    negativity, and objectivity.
114 #%%
115 breakdown = swn.senti_synset('breakdown.n.03')
116 print(breakdown)
117 print("Positive score = ", breakdown.pos_score())
118 print("Negative score = ", breakdown.neg_score())
119 print("Objective score = ", breakdown.obj_score())
120 #%%
121 sentence1 = ['I', 'had', 'a', 'breakdown', '.']
122 sentence2 = ['My', 'friend', 'had', 'a', 'breakdown'
    , 'before', 'her', 'exam', '.']
123
124 #%%
125
```