

WordNet

April 26, 2023

Summary of WordNet:

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.

First need to import wordNet:

```
[ ]: from nltk.corpus import wordnet as wn
     from nltk.wsd import lesk
     from nltk.corpus import sentiwordnet as swn
```

Select a noun and print out its synsets:

```
[ ]: wn.synsets("book")
```

Select a synset, and get its definition, usage examples, and lemmas:

```
[ ]: #definition
     wn.synset('book.n.01').definition()
```

```
[ ]: #examples
     wn.synset('book.n.01').examples()
```

```
[ ]: #lemmas
     wn.synset('book.n.01').lemmas()
```

For the selected synset traverse up the WordNet hierarchy:

```
[ ]: book = wn.synset('book.n.01')
     hyp = book.hypernyms()[0]
     top = wn.synset('entity.n.01')
     while hyp:
         print(hyp)
         if hyp == top:
             break
         if hyp.hypernyms():
             hyp = hyp.hypernyms()[0]
```

WordNet Organization for Nouns: 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.

Output the list of hypernyms, hyponyms, meronyms, holonyms, and antonyms for the noun:

```
[ ]: #hypernyms
book = wn.synset('book.n.01')
print("hypernyms: ", book.hypernyms())
```

```
[ ]: #hyponyms
book = wn.synset('book.n.01')
print("hyponyms: ", book.hyponyms())
```

```
[ ]: #meronyms
book = wn.synset('book.n.01')
print("meronyms: ", book.member_meronyms())
```

```
[ ]: #holonyms
book = wn.synset('book.n.01')
print("holonyms: ", book.part_holonyms())
```

```
[ ]: #antonyms
book = wn.synset('book.n.01')
print("antonyms: ", book.lemmas()[0].antonyms())
```

Select a verb and output all synsets:

```
[ ]: wn.synsets("ran")
```

Select a synset, and get its definition, usage examples, and lemmas:

```
[ ]: #definition
wn.synset('run.v.01').definition()
```

```
[ ]: #examples
wn.synset('run.v.01').examples()
```

```
[ ]: #lemmas
wn.synset('run.v.01').lemmas()
```

For the selected synset traverse up the WordNet hierarchy:

```
[ ]: run = wn.synset('run.v.01')
hyp = run.hypernyms()[0]
top = wn.synset('entity.n.01')
while hyp:
    print(hyp)
    if hyp == top:
        break
    if hyp.hypernyms():
```

```
hyp = hyp.hypernyms()[0]
```

WordNet Organization for Verbs: Unlike for nouns, there is no uniform top level synset. So, trying to traverse up the WordNet hierarchy doesn't really work.

```
[ ]: #use morphy  
wn.morphy('run', wn.VERB)
```

```
[ ]: #select two similar words:  
orange = wn.synset('orange.n.01')  
strawberry = wn.synset('strawberry.n.01')  
  
#Run Wu-Palmer  
wn.wup_similarity(orange, strawberry)
```

```
[ ]: #run lesk algorithm  
sent = ['I', 'won', 'money', 'playing', 'poker', '.']  
print(lesk(sent, 'poker', 'n'))
```

The Wu-Palmer metric shows that orange and strawberry have a lot of common ancestor words. A score of 0.75 is pretty high.

SentiWordNet: 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.

```
[ ]: breakdown = swn.senti_synset('breakdown.n.03')  
print(breakdown)  
print("Positive score = ", breakdown.pos_score())  
print("Negative score = ", breakdown.neg_score())  
print("Objective score = ", breakdown.obj_score())
```

```
[ ]: sentence1 = ['I', 'had', 'a', 'breakdown', '.']  
sentence2 = ['My', 'friend', 'had', 'a', 'breakdown', 'before', 'her', 'exam', '␣  
↪ '.']
```