

LAB 6

Name: Satwik Hosamani

SUID:378000248

Word:'chilly'

Results:

```
# or combine the synonyms/lemma names, definitions and examples
for synset in wn.synsets('chilly'):
    print (synset, ": ")
    print ('      ', synset.lemma_names())
    print ('      ', synset.definition())
    print ('      ', synset.examples())

Synset('chili.n.02') :
    ['chili', 'chili_pepper', 'chilli', 'chilly', 'chile']
    very hot and finely tapering pepper of special pungency
    []
Synset('chilly.s.01') :
    ['chilly']
    not characterized by emotion
    ['a female form in marble--a chilly but ideal medium for depicting abstract virtues"-C.W.Cunningham']
Synset('chilly.s.02') :
    ['chilly', 'parky']
    appreciably or disagreeably cold
    []
Synset('chilly.s.03') :
    ['chilly']
    lacking warmth of feeling
    ['a chilly greeting']
```

```
# trace paths of a synset by visiting its hypernyms
chilly1.hypernyms()

[Synset('hot_pepper.n.02')]

[19] # number of paths from the synset to the root concept "entity"
paths=Chilly1.hypernym_paths()
print(len(paths) )
# look at the first path
paths[0]

1
[Synset('entity.n.01'),
 Synset('physical_entity.n.01'),
 Synset('matter.n.03'),
 Synset('solid.n.01'),
 Synset('food.n.02'),
 Synset('produce.n.01'),
 Synset('vegetable.n.01'),
 Synset('solanaceous_vegetable.n.01'),
 Synset('pepper.n.04'),
 Synset('hot_pepper.n.02'),
 Synset('chili.n.02')]
```

```
▶ [synset.name() for synset in paths[0]]
```

```
↳ ['entity.n.01',  
    'physical_entity.n.01',  
    'matter.n.03',  
    'solid.n.01',  
    'food.n.02',  
    'produce.n.01',  
    'vegetable.n.01',  
    'solanaceous_vegetable.n.01',  
    'pepper.n.04',  
    'hot_pepper.n.02',  
    'chili.n.02']
```

```
print(spring.hypernym_paths())  
print(summer.hypernym_paths())  
print(winter.hypernym_paths())
```

```
[[Synset('entity.n.01'), Synset('abstraction.n.06'),  
Synset('measure.n.02'), Synset('fundamental_quantity.n.01'),  
Synset('time_period.n.01'), Synset('season.n.02'), Synset('spring.n.01')]]  
[[Synset('entity.n.01'), Synset('abstraction.n.06'),  
Synset('measure.n.02'), Synset('fundamental_quantity.n.01'),  
Synset('time_period.n.01'), Synset('season.n.02'), Synset('summer.n.01')]]  
[[Synset('entity.n.01'), Synset('abstraction.n.06'),  
Synset('measure.n.02'), Synset('fundamental_quantity.n.01'),  
Synset('time_period.n.01'), Synset('season.n.02'), Synset('winter.n.01')]]
```

```
[25] print(spring.lowest_common_hypernyms(summer))
      print(spring.lowest_common_hypernyms(winter))

[Synset('season.n.02')]
[Synset('season.n.02')]

[26] print(spring.min_depth() )
      print(wn.synset('season.n.02').min_depth() )
      print(wn.synset('entity.n.01').min_depth())

6
5
0

[27] # the path similarity gives a similarity score between 0 and 1
      print(spring.path_similarity(summer) )
      print(spring.path_similarity(winter))

0.3333333333333333
0.3333333333333333
```

```
# define 2 more words and look at their similarity
Rain = wn.synset('rain.n.01')
window = wn.synset('window.n.01')
# note the least ancestor of these two words
print(spring.lowest_common_hypernyms(Rain))
print(spring.lowest_common_hypernyms(window))

[Synset('entity.n.01')]
[Synset('entity.n.01')]

# Leacock-Chodorow Similarity, also uses path lengths and others
print(spring.lch_similarity(summer))
print(spring.lch_similarity(winter))
print(spring.lch_similarity(Rain))

2.538973871058276
2.538973871058276
0.8649974374866046
```

```
[33] # try Resnik Similarity
      print(spring.res_similarity(summer, brown_ic))
      print(spring.res_similarity(winter, brown_ic))
      print(spring.res_similarity(window, brown_ic))

7.904569247165815
7.904569247165815
-0.0
```

Sentiment Analysis of Words

```
[35] # sentiwordnet has the same synsets as wordnet, use wn functions
print(list(swn.senti_synsets('chilly')))
print(wn.synsets('chilly'))

[SentiSynset('chili.n.02'), SentiSynset('chilly.s.01'), SentiSynset('chilly.s.02'), SentiSynset('chilly.s.03')]
[Synset('chili.n.02'), Synset('chilly.s.01'), Synset('chilly.s.02'), Synset('chilly.s.03')]

# the print function gives the positive and negative scores
breakdown3 = swn.senti_synset('chili.n.02')
print (breakdown3)

<chili.n.02: PosScore=0.0 NegScore=0.0>

[37] # there are also separate functions for all the scores
print(breakdown3.pos_score())
print(breakdown3.neg_score())
print(breakdown3.obj_score())

0.0
0.0
1.0

[38] chillyswn1 = swn.senti_synset('chili.n.02')
print(chillyswn1)
print(chillyswn1.obj_score())

<chili.n.02: PosScore=0.0 NegScore=0.0>
1.0
```

Observations:

1. I took a word that has the same sound but two different spelling and meanings chilli pepper or chilly weather.
2. To see the similarity function I took a word window that is remotely related to weather and that was seen from the results, even with different types of the similarity measure.
3. The sentiment analysis since the word doesn't depict any emotion hence they have an objective score of 1.0

Lessons Learned:

1. WordNet can be used to investigate the words.
2. Sysnets can be used as an identifier for the words.
3. Finding similarity between words
4. Sentiment analysis of words.