Santhosh.S

225229133

Natural Language Processing Lab

Lab 14: Word Sense Disambiguation with Improved Lesk Algorithm

Exercise-1

Lesk algorithms syntax:

lesk(context_sentence,ambiguous_word,pos=None,synsets=None)

In [1]:
```
import nltk
from nltk.wsd import lesk
from nltk.corpus import wordnet as wn
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Out[1]:
```
True
```

In [2]:
```
for ss in wn.synsets('bass'):
    print(ss,ss.definition())
```

```
Synset('bass.n.01') the lowest part of the musical range
Synset('bass.n.02') the lowest part in polyphonic music
Synset('bass.n.03') an adult male singer with the lowest voice
Synset('sea_bass.n.01') the lean flesh of a saltwater fish of the family Serr
anidae
Synset('freshwater_bass.n.01') any of various North American freshwater fish
with lean flesh (especially of the genus Micropterus)
Synset('bass.n.06') the lowest adult male singing voice
Synset('bass.n.07') the member with the lowest range of a family of musical i
nstruments
Synset('bass.n.08') nontechnical name for any of numerous edible marine and f
reshwater spiny-finned fishes
Synset('bass.s.01') having or denoting a low vocal or instrumental range
```

In [3]:
```
print(lesk('I went fishing for some sea bass'.split(),'bass','n'))
```

```
Synset('bass.n.08')
```

In [4]:
```
print(lesk('The bass line of the song is too weak'.split(),'bass','s'))
```

```
Synset('bass.s.01')
```

In [5]:
```
print(lesk('Avishai Cohen is an Israeli jazz musician. He plays double bass and
           'bass',pos='n'))
```

```
Synset('sea_bass.n.01')
```

Exercise-2

In [6]:
```
for ss in wn.synsets('chair'):
    print(ss,ss.definition())
```

```
Synset('chair.n.01') a seat for one person, with a support for the back
Synset('professorship.n.01') the position of professor
Synset('president.n.04') the officer who presides at the meetings of an organ
ization
Synset('electric_chair.n.01') an instrument of execution by electrocution; re
sembles an ordinary seat for one person
Synset('chair.n.05') a particular seat in an orchestra
Synset('chair.v.01') act or preside as chair, as of an academic department in
a university
Synset('moderate.v.01') preside over
```

In [7]:
```
syn = wn.synsets('chair')[0]
print(syn)
```

```
Synset('chair.n.01')
```

In [8]:
```
print ("Synset name : ", syn.name())

print ("Synset hypernym : ", syn.hypernyms())
```

```
Synset name :  chair.n.01
Synset hypernym :  [Synset('seat.n.03')]
```

In [9]:
```
print ("Synset  hyper-hypernym : ", syn.root_hypernyms)
```

```
Synset  hyper-hypernym :  <bound method Synset.root_hypernyms of Synset('chai
r.n.01')>
```

Exercise-3

In [10]:
```
from nltk.corpus import wordnet as wn
from nltk.stem import PorterStemmer
from itertools import chain
```

In [11]:
```
bank_sents=['I went to the bank to deposit my money','The river bank was full
plant_sents = ['The workers at the industrial plant were overworked','The plan
ps = PorterStemmer()
```

In [12]:
```
def my_lesk(context_sentence, ambiguous_word,pos=None,
            stem=True, hyperhypo=True):
    max_overlaps = 0
    lesk_sense = None
    context_sentence = context_sentence.split()
    for ss in wn.synsets(ambiguous_word):
     # If POS is specified.
        if pos and ss.pos is not pos:
            continue
        lesk_dictionary = []

     # Includes definition.
        defns = ss.definition().split()
        lesk_dictionary += defns

     # Includes lemma_names.
        lesk_dictionary += ss.lemma_names()

     # Optional: includes lemma_names of hypernyms and hyponyms.
        if hyperhypo == True:
            hhwords = ss.hypernyms() + ss.hyponyms()
            lesk_dictionary += list(chain(*[w.lemma_names() for w in hhwords]

     # Matching exact words causes sparsity, so lets match stems.
        if stem == True:
            lesk_dictionary = [ps.stem(w) for w in lesk_dictionary]
            context_sentence = [ps.stem(w) for w in context_sentence]
        overlaps = set(lesk_dictionary).intersection(context_sentence)

        if len(overlaps) > max_overlaps:
            lesk_sense = ss
            max_overlaps = len(overlaps)
    return lesk_sense
```

In [13]:
```
# evaluate senses

print("Context:", bank_sents[0])
answer = my_lesk(bank_sents[0],'bank')
print("Sense:", answer)
print("Definition:",answer.definition)
```

```
Context: I went to the bank to deposit my money
Sense: Synset('depository_financial_institution.n.01')
Definition: <bound method Synset.definition of Synset('depository_financial_i
nstitution.n.01')>
```

In [14]:
```
print("Context:", bank_sents[1])
answer = my_lesk(bank_sents[1],'bank')
print("Sense:", answer)
print("Definition:",answer.definition)
```

```
Context: The river bank was full of dead fishes
Sense: Synset('bank.n.01')
Definition: <bound method Synset.definition of Synset('bank.n.01')>
```

In [15]:
```
print("Context:", plant_sents[0])
answer = my_lesk(plant_sents[0],'bank')
print("Sense:", answer)
print("Definition:",answer.definition)
```

```
Context: The workers at the industrial plant were overworked
Sense: Synset('savings_bank.n.02')
Definition: <bound method Synset.definition of Synset('savings_bank.n.02')>
```

Exercise-4

Learn further examples for synsets at

https://www.programcreek.com/python/example/91604/nltk.corpus.wordnet.synsets (https://www.programcreek.com/python/example/91604/nltk.corpus.wordnet.synsets)