自然语言处理——wordnet语料库的使用，判断是否存在共指指代

一. 使用nltk中的wordnet语料库

1.找出以下单词的同义词集、查看同义词集中的所有单词、查看同义词的具体定义及例子：dog, apple, fly

```
import nltk
from nltk.corpus import wordnet as wn

# 获取一个同义词集的所有单词
def get_lemma(synset_name):
    synsets = wn.synsets(synset_name)
    for i in range(len(synsets)):
        l_name = synsets[i].lemma_names()
        print(synsets[i],'如下所示 : ')
        print(l_name)

# 获取同义词的定义
def get_def(synset_name):
    synsets = wn.synsets(synset_name)
    for i in range(len(synsets)):
        l_name = synsets[i].definition()
        print(synsets[i],'如下所示 : ')
        print(l_name)

# 获取同义词的例子
def get_example(synset_name):
    synsets = wn.synsets(synset_name)
    for i in range(len(synsets)):
        l_name = synsets[i].examples()
        print(synsets[i],'如下所示 : ')
        print(l_name)

# 练习1.1
list = ['dog', 'apple', 'fly']
for item in list:
    print('1. '+item+'同义词集如下\n')
    print(wn.synsets(item))
    print()
    print('2. '+item+'同义词集中的所有单词如下\n')
    get_lemma(item)
    print()
    print('3. 同义词的具体定义\n')
    get_def(item)
    print()
```

```
    print('4. 同义词的具体例子\n')
    get_example(item)
    print('\n------------------------------\n')
```

实验结果：

这里只展示dog的，其他输出也是类似，由于篇幅就不截图显示

- 同义词集 `.synsets()`
- 同义词集中的所有单词 `.lemma_names()`
- 同义词具体定义 `.definition()`
- 同义词的具体例子 `.examples()`

```
λ python wordnet_test.py
1. dog同义词集如下

[Synset('dog.n.01'), Synset('frump.n.01'), Synset('dog.n.03'), Synset('cad.n.01'), Synset('frank.n.02'), Synset('pawl.n.01'), Synset('andiron.n.01'), Synset('chase.v.01')]

2. dog同义词集中的所有单词如下

Synset('dog.n.01') 如下所示 :
['dog', 'domestic_dog', 'Canis_familiaris']
Synset('frump.n.01') 如下所示 :
['frump', 'dog']
Synset('dog.n.03') 如下所示 :
['dog']
Synset('cad.n.01') 如下所示 :
['cad', 'bounder', 'blackguard', 'dog', 'hound', 'heel']
Synset('frank.n.02') 如下所示 :
['frank', 'frankfurter', 'hotdog', 'hot_dog', 'dog', 'wiener', 'wienerwurst', 'weenie']
Synset('pawl.n.01') 如下所示 :
['pawl', 'detent', 'click', 'dog']
Synset('andiron.n.01') 如下所示 :
['andiron', 'firedog', 'dog', 'dog-iron']
Synset('chase.v.01') 如下所示 :
['chase', 'chase_after', 'trail', 'tail', 'tag', 'give_chase', 'dog', 'go_after', 'track']

3. 同义词的具体定义

Synset('dog.n.01') 如下所示 :
a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds
Synset('frump.n.01') 如下所示 :
a dull unattractive unpleasant girl or woman
Synset('dog.n.03') 如下所示 :
informal term for a man
Synset('cad.n.01') 如下所示 :
someone who is morally reprehensible
Synset('frank.n.02') 如下所示 :
a smooth-textured sausage of minced beef or pork usually smoked; often served on a bread roll
Synset('pawl.n.01') 如下所示 :
a hinged catch that fits into a notch of a ratchet to move a wheel forward or prevent it from moving backward
Synset('andiron.n.01') 如下所示 :
metal supports for logs in a fireplace
Synset('chase.v.01') 如下所示 :
go after with the intent to catch

4. 同义词的具体例子
```

```
4. 同义词的具体例子

Synset('dog.n.01') 如下所示 :
['the dog barked all night']
Synset('frump.n.01') 如下所示 :
['she got a reputation as a frump', "she's a real dog"]
Synset('dog.n.03') 如下所示 :
['you lucky dog']
Synset('cad.n.01') 如下所示 :
['you dirty dog']
Synset('frank.n.02') 如下所示 :
[]
Synset('pawl.n.01') 如下所示 :
[]
Synset('andiron.n.01') 如下所示 :
['the andirons were too hot to touch']
Synset('chase.v.01') 如下所示 :
['The policeman chased the mugger down the alley', 'the dog chased the rabbit']

------------------------------
```

2. 查看以下单词对的语义相似度：good, beautiful；good,

bad; dog, cat

```
def get_similarity(w1, w2):
    s1 = wn.synsets(w1)
    s2 = wn.synsets(w2)
    sim_max = 0
    for s1_item in s1:
        for s2_item in s2:
            sim = s1_item.path_similarity(s2_item)
            if (sim is None):
                sim = 0
            sim_max = max(sim_max, sim)
    return sim_max

# 练习1.2
print('查看good与beautiful的语义相似度 : ')
print(get_similarity('good', 'beautiful'))
print()
print('查看good与bad的语义相似度 : ')
print(get_similarity('good', 'bad'))
print()
print('查看dog与cat的语义相似度 : ')
print(get_similarity('dog', 'cat'))
print('\n------------------------------\n')
```

实验结果:

```
------------------------------

查看good与beautiful的语义相似度 :
0

查看good与bad的语义相似度 :
0.3333333333333333

查看dog与cat的语义相似度 :
0.2

------------------------------
```

3. 找出以下单词的蕴含(entailments)关系和反义词：walk,

supply, hot

```
def get_entailments(w):
    s = wn.synsets(w)
```

```
    for item in s:
        en = item.entailments()
        if len(en) > 0:
            print(item,' : ')
            print(en)


def get_antonyms(w):
    s = wn.synsets(w)
    for item in s:
        lms = item.lemmas()
        for l in lms:
            a = l.antonyms()
            if len(a) > 0:
                print(item,' : ')
                print(a)


list2 = ['walk', 'supply', 'hot']

 # 练习1.3
for w in list2:
    print(w + '的蕴含关系 : ')
    get_entailments(w)
    print()
    print(w + '的反义词 :  ')
    get_antonyms(w)
    print('\n----------------------------\n')
```

实验结果：

二. 判断下列句子中是否存在的共指指代，有的话找出共指链

使用工具 https://github.com/huggingface/neuralcoref

- My sister has a dog. She loves him.
- Some like to play football, others are fond of basketball.
- The more a man knows, the more he feels his ignorance

```
import spacy
nlp = spacy.load('en_core_web_sm')

# Add neural coref to SpaCy's pipe
import neuralcoref
neuralcoref.add_to_pipe(nlp)

l = ['My sister has a dog. She loves him.',
'Some like to play football, others are fond of basketball.',
```

```
'The more a man knows, the more he feels his ignorance']

for item in l:
    doc = nlp(item)
    flag = doc._.has_coref
    print('\n' + item + ' 如下所示')
    print('判断是否存在共指指代 ： ')
    print(flag)
    print('共指链 ： ')
    if flag:
        print(doc._.coref_clusters)
```

实验结果：

```
λ  python nlp_test.py

My sister has a dog. She loves him. 如下所示
判断是否存在共指指代 ：
True
共指链 ：
[My sister: [My sister, She], a dog: [a dog, him]]

Some like to play football, others are fond of basketball. 如下所示
判断是否存在共指指代 ：
False
共指链 ：

The more a man knows, the more he feels his ignorance 如下所示
判断是否存在共指指代 ：
True
共指链 ：
[The: [The, the], a man: [a man, he, his]]
```