# 第二章

28.**使用预定义的相似性度量给下面每个词对的相似性打分。按相似性递减的顺序排名。你的排名与这里给出的顺序有多接近？由Miller&Charles(1998)实验得出的顺序：car-automobile、gem-jewel、journey-voyage、boy-lad、coast-shore、asylum-madhouse、magician-wizard 、midday-noon、furnace-stove、food-fruit、bird-cock、bird-crane、tool-implement、brother-monk、lad-brother、crane-implement、journey-car、monk-oracle、cemetery-woodland、food-rooster、coast-hill、forest-graveyard、shore-woodland、monk-slave、coast-forest、lad-wizard、chord-smile、glass-magician、rooster-voyage、noon-string**

**from** nltk.corpus **import** wordnet **as** wn  
pairs = {**'car'**:**'automobile'**,**'gem'**:**'jewel'**,**'voyage'**:**'journey'**,**'boy'**:**'lad'**,**'shore'**:**'coast'**,**'asylum'**:**'madhouse'**,**'magician'**:**'wizard'**,**'midday'**:**'noon'**,**'furnace'**:**'stove'**,**'fruit'**:**'food'**,**'bird'**:**'cock'**,**'crane'**:**'bird'**,**'tool'**:**'implement'**,**'brother'**:**'monk'**,**'lad'**:**'brother'**,**'implement'**:**'crane'**,**'journey'**:**'car'**,**'oracle'**:**'monk'**,**'cemetery'**:**'woodland'**,**'food'**:**'rooster'**,**'hill'**:**'coast'**,**'forest'**:**'graveyard'**,**'woodland'**:**'shore'**,**'monk'**:**'slave'**,**'coast'**:**'forest'**,**'wizard'**:**'lad'**,**'chord'**:**'smile'**,**'glass'**:**'magician'**,**'rooster'**:**'voyage'**,**'noon'**:**'string'**}

List1=[]  
List2=[]  
**for** key **in** pairs:  
 word1 = wn.synset(str(key) + **'.n.01'**)  
 word2 = wn.synset(str(pairs[key]) + **'.n.01'**)  
 similarity = word1.path\_similarity(word2)  
 List1.append(**'%.4f'** %similarity)  
 List2.append(key + **'-'** + pairs[key])

**print** List1

['0.0909', '0.2000', '0.2000', '0.1250', '0.2500', '0.0588', '0.2000', '0.1111', '0.0625', '0.0417', '0.0714', '0.0769', '0.0625', '1.0000', '0.2000', '0.1429', '0.5000', '0.0909', '0.1111', '0.0909', '0.1111', '0.1667', '0.1250', '0.3333', '1.0000', '0.1250', '0.5000', '0.0500', '0.1250', '0.1000']

**print** List2

['chord-smile', 'woodland-shore', 'monk-slave', 'asylum-madhouse', 'voyage-journey', 'noon-string', 'wizard-lad', 'cemetery-woodland', 'food-rooster', 'rooster-voyage', 'forest-graveyard', 'furnace-stove', 'bird-cock', 'midday-noon', 'hill-coast', 'lad-brother', 'tool-implement', 'coast-forest', 'glass-magician', 'fruit-food', 'crane-bird', 'magician-wizard', 'gem-jewel', 'boy-lad', 'car-automobile', 'brother-monk', 'shore-coast', 'journey-car', 'oracle-monk', 'implement-crane']

p\_dict={}  
**for** i **in** range(len(List1)):  
 p\_dict[List2[i]]=List1[i]  
p\_dict=sorted(p\_dict.items(),key=**lambda** x:x[1],reverse=True)  
  
**print** p\_dict

[('midday-noon', '1.0000'), ('car-automobile', '1.0000'), ('shore-coast', '0.5000'), ('tool-implement', '0.5000'), ('boy-lad', '0.3333'), ('voyage-journey', '0.2500'), ('hill-coast', '0.2000'), ('woodland-shore', '0.2000'), ('monk-slave', '0.2000'), ('wizard-lad', '0.2000'), ('magician-wizard', '0.1667'), ('lad-brother', '0.1429'), ('brother-monk', '0.1250'), ('asylum-madhouse', '0.1250'), ('oracle-monk', '0.1250'), ('gem-jewel', '0.1250'), ('cemetery-woodland', '0.1111'), ('glass-magician', '0.1111'), ('crane-bird', '0.1111'), ('implement-crane', '0.1000'), ('fruit-food', '0.0909'), ('chord-smile', '0.0909'), ('coast-forest', '0.0909'), ('furnace-stove', '0.0769'), ('forest-graveyard', '0.0714'), ('bird-cock', '0.0625'), ('food-rooster', '0.0625'), ('noon-string', '0.0588'), ('journey-car', '0.0500'), ('rooster-voyage', '0.0417')]

下面这个表两种方法的对比。

左侧是使用预定义的相似性度量给词对的相似性打的分排序后得出的结果。

右侧是Miller&Charles(1998)实验得出的顺序。

两者顺序有一定差异。

