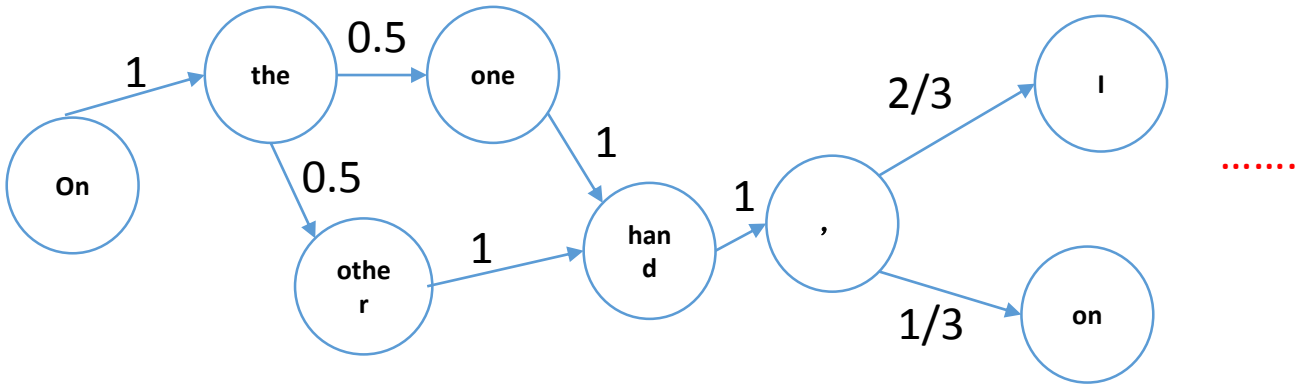
Stochastic Signal Processing

Experiment 4

Markov chain experiment

On the one hand, I really want to go, on the other hand, I have to look after my parents.



Count occurrences of each word

On : {the: 2}

the : {one : 1, other :1}

one: {hand:1}

hand:{, : 2}

, :{I : 2, on: 1}

Now, if the first word is ‘the’,

the probability of occurrence of ‘one’ is 50% (1/2), and ‘other’ is 50% (1/2).

If the word is ‘,’ ,

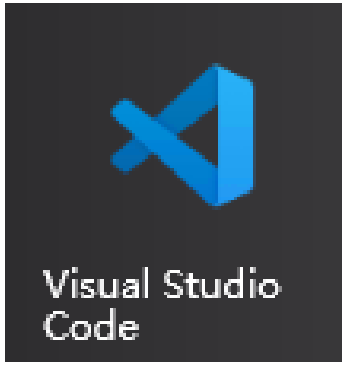
The probability of occurrence of ‘I’ is 66.67% (2/3),

and ‘on’ is 33.33% (1/3).

…… future outcomes based solely on its present state

Markov chain experiment

1. Pycharm or Vscode IDE



2. Google Colaboratory (Web Version)

Python - dictionary

Array = [‘a’, ‘b’, ‘c’]

wordList ={key1 : value1, key2 : value2 …….};

For example:

```
>>>namedict = {'Name1': 'Tony', 'Name2': 'Jenny', 'Name3': 'Curry'}

>>>print(namedict['Name2'])

Jenny

>>>  agedict  = {'Tony': 15,  'Jenny': 19, 'Curry': 30}
>>>  print(agedict['Tony'])
15

>>>print("agedict[{}]:{}".format(namedict['Name2'], agedict[namedict['Name2']]))

agedict[Jenny]:19
```

```
>>>for name,age in agedict.items():
>>>     print(name, '>>',age)
```

Tony >> 15

Jenny >>19

Curry >>30

Python – dictionary_2D

```
Array_2d = [[‘a1’, ‘b1’, ‘c1’],
            [‘a2’, ‘b2’, ‘c2’],
            [‘a3’, ‘b3’, ‘c3’]]
wordList_2d = {'China': {'Guangzhou': 4000, 'Shenzhen': 5000},
               'Amercia': {'Los Angeles': 2000,'New York':3000}}
print(wordList_2d ['China']['Guangzhou'])
wordList_2d ['China']['Shenzhen']= 3500
wordList_2d ['China']['Beijing']= 4000
print(wordList_2d)
for country, city in wordList_2d.items():
    for name, value in city.items():
        print(name, '>>', value)
```

4000

{'China': {'Guangzhou': 4000, 'Shenzhen': 3500, 'Beijing': 4000},
'Amercia': {'Los Angeles': 2000, 'New York': 3000}}

Guangzhou>>4000

Shenzhen>>3500

Beijing>>4000

Los Angeles>>2000

New York>>3000

Python – dictionary_2D

```
Array_2d = [[‘a1’, ‘b1’, ‘c1’],
            [‘a2’, ‘b2’, ‘c2’],
            [‘a3’, ‘b3’, ‘c3’]]
wordList_2d = {'China': {'Guangzhou': 4000, 'Shenzhen': 5000},
               'Amercia': {'Los Angeles': 2000,'New York':3000}}
print(wordList_2d ['China']['Guangzhou'])
wordList_2d ['China']['Shenzhen']= 3500
wordList_2d ['China']['Beijing']= 4000
print(wordList_2d)
for country, city in wordList_2d.items():
    for name, value in city.items():
        print(name, '>>', value)
```

4000

```
{'China': {'Guangzhou': 4000, 'Shenzhen': 3500,
'Beijing': 4000},
 'Amercia': {'Los Angeles': 2000, 'New York': 3000}}
Guangzhou>>4000
Shenzhen>>3500
Beijing>>4000
Los Angeles>>2000
New York>>3000
```

Experiment – Markov chain

Experimental Report 4

1、Use a piece of known text to generate a random short text of 100 words using the knowledge of Markov chains. Based on the code provided, write a flowchart and understandings/comments of this code.

2、Because punctuation mark plays a key role in sentence breaking, its frequency weight should be three times that of words. Please modify the source code to meet this requirement.(Just modify the code and submit it, no need to write it in experiment report)

URL：https://blog.csdn.net/Freyua_xx/article/details/121747591

Format requirements

- send email to 刘译哲 before 23:59:59, 29/06
  - One experimental report file in .doc or .docx or .pdf format and one modified code file are required to submitted.
- this experiment is only one example of text generation using the basic Markov model, only understanding is required, thus
  - you are required to understand the basic idea, and try to run the python code only
  - and show your understanding in your report
  - the experimental report 4 will take approx. 3 out of the 100 points of the whole course.