Technology Science Information Networks Computing

Lecturer: Ting Wang (王挺)

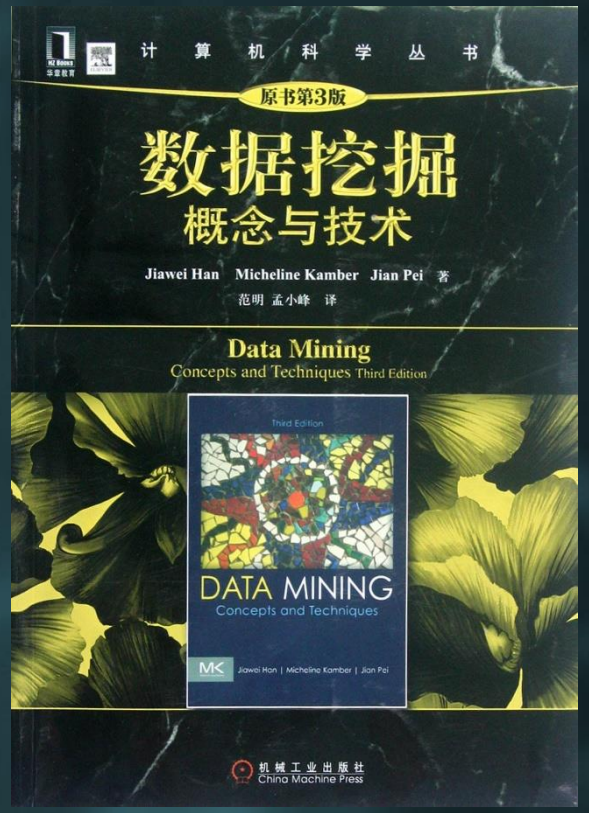
利物浦大学计算机博士

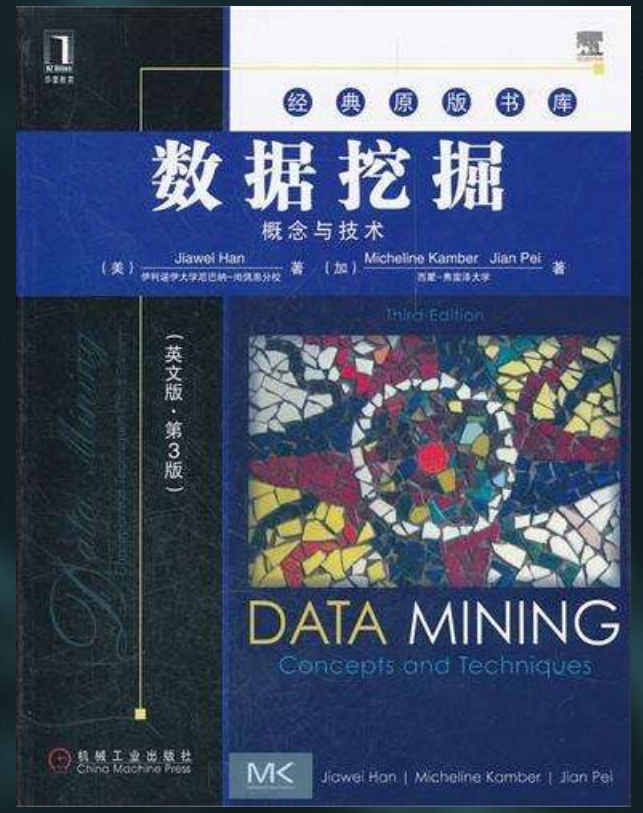
清华大学计算机博士后

电子信息技术高级工程师

上海外国语大学网络与新媒体副教授

浙江清华长三角研究院海纳认知与智能研究中心主任





Chapter 6

Mining Frequent Patterns, Association and Correlations

Chapter 6: Mining Frequent Patterns, Association and Correlations

1. Frequent pattern

a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set

2. Frequent itemsets

A set of items is referred to as an **itemset**. An itemset that contains *k* items is a ***k*-itemset**. If the relative support of an itemset *I* satisfies a prespecified **minimum support threshold** (i.e., the absolute support of *I* satisfies the corresponding **minimum support count threshold**), then *I* is a **frequent** itemset.

频繁项集：经常出现在一块的物品的集合

Chapter 6: Mining Frequent Patterns, Association and Correlations

3. Association Rules

Find all the rules $X \rightarrow Y$(s,c) with minimum support and confidence

s: support （支持度） $(X \rightarrow Y) = P(X \cup Y) = support(X \cup Y)$

c: confidence （置信度） $(X \rightarrow Y) = P(Y \mid X) = \frac{support(X \cup Y)}{support(X)}$

关联规则：暗示两种物品之间可能存在很强的关系

同时满足**最小支持度阈值**和**最小置信度阈值**的规则称为强规则
一般而言，关联规则的挖掘分为两步：

1. 找出所有频繁项集，即候选规则
2. 对所有候选规则计算置信度，找出其中的强规则

Chapter 6: Mining Frequent Patterns, Association and Correlations

4. Super-patterns（超模式）/ Super-itemset（超集）

An itemset X is a super-pattern of Y if every element in Y is also in X

5. Close-patterns（封闭模式）/ Closed frequent itemset（闭频繁项集）

An itemset X is **closed** if X is *frequent* and there exists *no super-pattern* Y ↄ X, *with the same support* as X

6. Max-patterns（最大模式）/ Maximal Frequent Itemset（最大频繁项集）

An itemset X is a max-pattern if X is frequent and there exists no frequent super-pattern Y ↄ X

Chapter 6: Mining Frequent Patterns, Association and Correlations

6. Apriori

Steps:

① 首先会生成所有单个元素集合的项集列表；

② 扫描记录来查看哪些项集满足最小支持度要求，那些不满足最小支持度的集合会被去掉；

③ 对剩下的集合进行组合以生成包含两个元素的项集；

④ 接下来重新扫描交易记录，去掉不满足最小支持度的项集，重复进行直到所有项集都被去掉。

Chapter 6: Mining Frequent Patterns, Association and Correlations

7. FPGrowth

Steps:

① 遍历一次数据集，统计每个元素出现的次数，并对每个元素按出现次数重排序，然后根据最小支持度把出现次数较小的元素滤掉；

② 构造FP树。从根节点∅开始，将过滤并排序后的样本一个个加入树中，若FP树不存在现有元素则添加分支，若存在则增加相应的值；

③ 根据FP树，找到频繁项。

Chapter 6: Mining Frequent Patterns, Association and Correlations

8. Correlation and Association

Correlation

	M	O	N	K	E	Y	D	A	U	C	H	I
MONKEY	1	1	1	1	1	1	0	0	0	0	0	0
DONKEY	0	1	1	1	1	1	1	0	0	0	0	0
MAKE	1	0	0	1	1	0	0	1	0	0	0	0
MUCHY	1	0	0	0	0	1	0	0	1	1	1	0
COOKIE	0	1	0	1	1	0	0	0	0	1	0	1

	M	O	N	K	E	Y	D	A	U	C	H	I
M	1	-0.66667	-0.16667	-0.40825	-0.40825	0.166667	-0.61237	0.408248	0.408248	-0.16667	0.408248	-0.61237
O	-0.66667	1	0.666667	0.612372	0.612372	0.166667	0.408248	-0.61237	-0.61237	-0.16667	-0.61237	0.408248
N	-0.16667	0.666667	1	0.408248	0.408248	0.666667	0.612372	-0.40825	-0.40825	-0.66667	-0.40825	-0.40825
K	-0.40825	0.612372	0.408248	1	1	-0.40825	0.25	0.25	-1	-0.61237	-1	0.25
E	-0.40825	0.612372	0.408248	1	1	-0.40825	0.25	0.25	-1	-0.61237	-1	0.25
Y	0.166667	0.166667	0.666667	-0.40825	-0.40825	1	0.408248	-0.61237	0.408248	-0.16667	0.408248	-0.61237
D	-0.61237	0.408248	0.612372	0.25	0.25	0.408248	1	-0.25	-0.25	-0.40825	-0.25	-0.25
A	0.408248	-0.61237	-0.40825	0.25	0.25	-0.61237	-0.25	1	-0.25	-0.40825	-0.25	-0.25
U	0.408248	-0.61237	-0.40825	-1	-1	0.408248	-0.25	-0.25	1	0.612372	1	-0.25
C	-0.16667	-0.16667	-0.66667	-0.61237	-0.61237	-0.16667	-0.40825	-0.40825	0.612372	1	0.612372	0.612372
H	0.408248	-0.61237	-0.40825	-1	-1	0.408248	-0.25	-0.25	1	0.612372	1	-0.25
I	-0.61237	0.408248	-0.40825	0.25	0.25	-0.61237	-0.25	-0.25	-0.25	0.612372	-0.25	1

Association

```
import pyfpgrowth

transactions = [['M', 'O', 'N', 'K', 'E', 'Y'],
                ['D', 'O', 'N', 'K', 'E', 'Y'],
                ['M', 'A', 'K', 'E'],
                ['M', 'U', 'C', 'K', 'Y'],
                ['C', 'O', 'O', 'K', 'I', 'E']]
patterns = pyfpgrowth.find_frequent_patterns(transactions, 3)
rules = pyfpgrowth.generate_association_rules(patterns, 0.8)
print(patterns)
print(rules)
```

```
Run: testfpgrowth1
"C:\Program Files\Python36\python.exe"
 D:/Research/Projects/NewsEventDetection/testfpgrowth1.py
{('M',): 3, ('K', 'M'): 3, ('Y',): 3, ('K', 'Y'): 3, ('O',): 4,
 ('K', 'O'): 4, ('E', 'O'): 4, ('E', 'K'): 4, ('E', 'K', 'O'): 4,
 ('K',): 5}
{('M',): (('K',), 1.0), ('Y',): (('K',), 1.0), ('K',): (('E', 'O'),
  0.8), ('O',): (('E', 'K'), 1.0), ('E', 'K'): (('O',), 1.0), ('E',
  'O'): (('K',), 1.0), ('K', 'O'): (('E',), 1.0)}

Process finished with exit code 0
```

Chapter 6: Mining Frequent Patterns, Association and Correlations

Chapter 6: Mining Frequent Patterns, Association and Correlations

Headline is important!

Headline should be the biggest news in one day!

Question: How to make a headline?

Answer: We can employ fpgrowth algorithm to cope with this problem.

Chapter 6: Mining Frequent Patterns, Association and Correlations

Example:

What should be the headline of January 30, 2020’s international news?

Data:

We collect all the title of international news articles of January 30, 2020. Totally, 183.

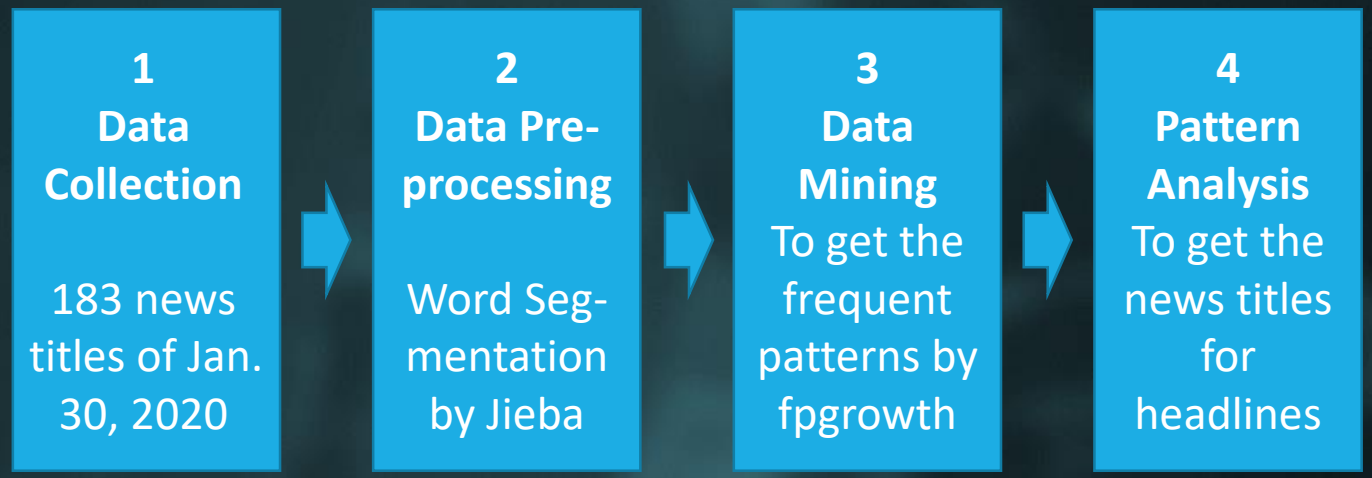
Language: All the articles are in Chinese.

Question:

How to get the most frequently mentioned news content?

Chapter 6: Mining Frequent Patterns, Association and Correlations

Steps:



Chapter 6: Mining Frequent Patterns, Association and Correlations

Result Analysis

```
# 使用分词器将list of files/文档 进行分词
# totalvocab_tokenized = []
transactions=[]
for i in news:
    allwords_tokenized = segment(i, "userdict2.txt", 'stopwords.txt')
    transactions.append(allwords_tokenized)
    print(allwords_tokenized)

#fpgrowth to fine frequency patterns
patterns = pyfpgrowth.find_frequent_patterns(transactions, 4)
patterns = dict((key, value) for key, value in patterns.items() if len(key)>3)
#output
npatterns=sorted(patterns.items(), key=lambda d:d[1], reverse = False)
print(npatterns)
print(list(npatterns)[-10:])
```

```
test1
['英国', '女画家', '露西', '普拉特', 'LucyPratt', '丰富多彩', '意象']
['昨夜', '架飞机', '改变', '航线', '直飞', '武汉']
[(('感染', '新型', '日本', '肺炎'), 4), (('冠状病毒', '感染', '日本', '肺炎'), 4), (('冠状病毒', '感染', '新型', '日本'), 4), (('冠状病毒', '感染', '新型', '日本', '肺炎'), 4), (('冠状病毒', '新型', '日本', '肺炎'), 4), (('冠状病毒', '感染', '新型', '肺炎'), 7)]
[(('感染', '新型', '日本', '肺炎'), 4), (('冠状病毒', '感染', '日本', '肺炎'), 4), (('冠状病毒', '感染', '新型', '日本'), 4), (('冠状病毒', '感染', '新型', '日本', '肺炎'), 4), (('冠状病毒', '新型', '日本', '肺炎'), 4), (('冠状病毒', '感染', '新型', '肺炎'), 7)]

Process finished with exit code 0
```

The frequency about 2019 coronavirus is 7, and there are 5 different patterns about 2019 coronavirus in Japan. Their frequency about 2019 Coronavirus are 4.

This indicates that the coronavirus in Japan can be a candidate of Jan. 30's headline.

Chapter 6: Mining Frequent Patterns, Association and Correlations

Result Analysis

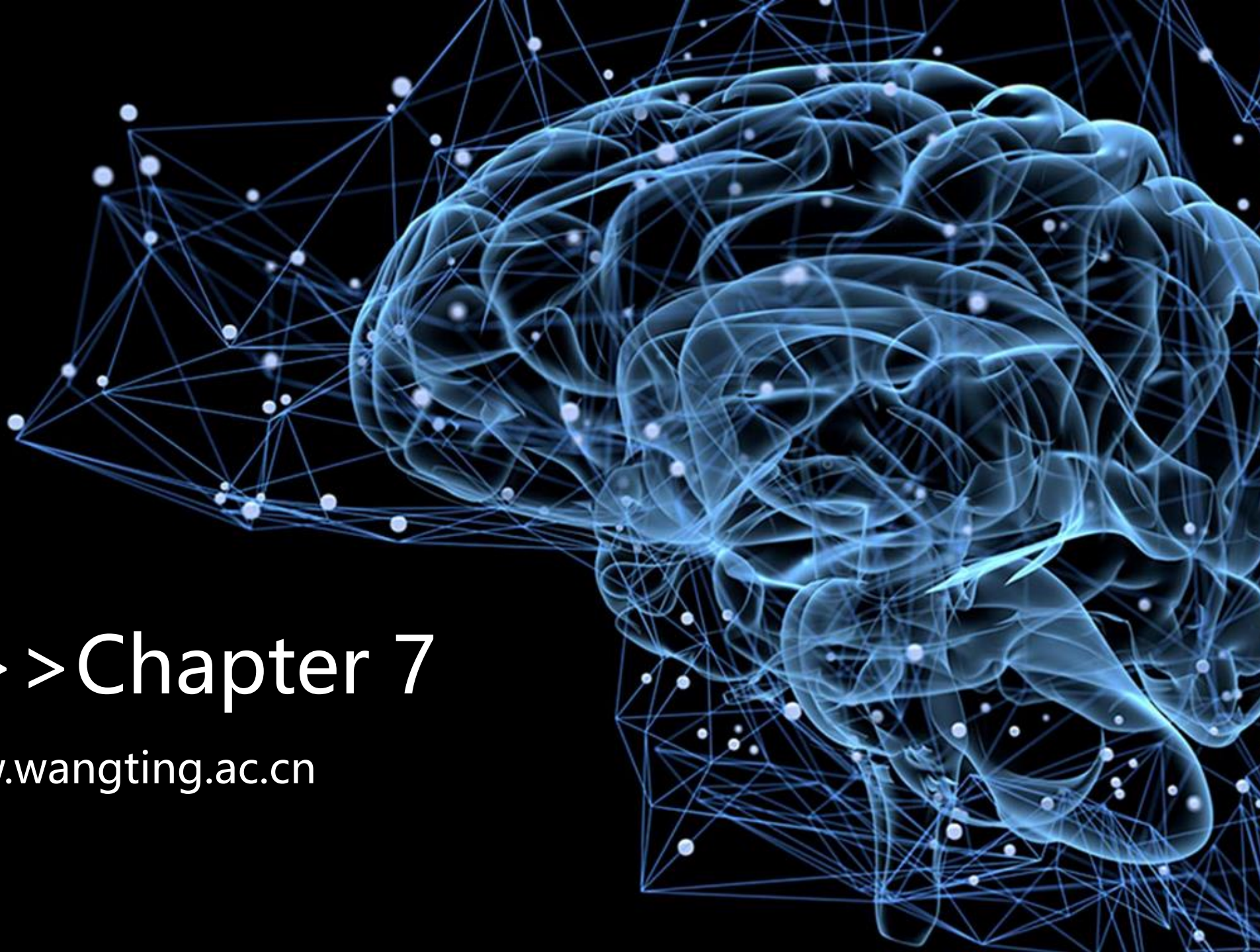
```
transactions=[]
for i in news:
    allwords_tokenized = segment(i, "userdict2.txt", 'stopwords.txt')
    transactions.append(allwords_tokenized)
    print(allwords_tokenized)

#fpgrowth to fine frequency patterns
patterns = pyfpgrowth.find_frequent_patterns(transactions, 2)
patterns = dict((key, value) for key, value in patterns.items() if len(key)>5)
#output
npatterns=sorted(patterns.items(), key=lambda d:d[1], reverse = False)
print(npatterns)
print(list(npatterns)[-10:])
```

test1

```
['庆阳市', '高速公路', '收费站', '公告']
['英国', '女画家', '露西', '普拉特', 'LucyPratt', '丰富多彩', '意象']
['昨夜', '架飞机', '改变', '航线', '直飞', '武汉']
[(('亿万', '女儿', '女婿', '比尔', '盖茨', '订婚'), 2), (('冠状病毒', '感染', '新型', '日本', '肺炎', '驻'), 2), (('中国', '冠状病毒', '感染', '新型', '肺炎', '驻'), 2), (('中国', '军机', '医疗', '派', '物资', '白俄罗斯'), 2), (('中', '人员', '包机', '发烧', '第二批', '返日'), 2), (('华人', '华侨', '抗击', '疫情', '病毒', '隔离'), 2), (('例', '冠状病毒', '感染', '新型', '病例', '肺炎'), 2), (('冠状病毒', '感染', '新型', '日本', '疫情', '肺炎'), 2)]
(('亿万', '女儿', '女婿', '比尔', '盖茨', '订婚'), 2), (('冠状病毒', '感染', '新型', '日本', '肺炎', '驻'), 2), (('中国', '冠状病毒', '感染', '新型', '肺炎', '驻'), 2), (('中国', '军机', '医疗', '派', '物资', '白俄罗斯'), 2), (('中', '人员', '包机', '发烧', '第二批', '返日'), 2), (('华人', '华侨', '抗击', '疫情', '病毒', '隔离'), 2), (('例', '冠状病毒', '感染', '新型', '病例', '肺炎'), 2), (('冠状病毒', '感染', '新型', '日本', '疫情', '肺炎'), 2)]
```

If the parameters are changed, more secondary news will appear.



Next>>Chapter 7

www.wangting.ac.cn