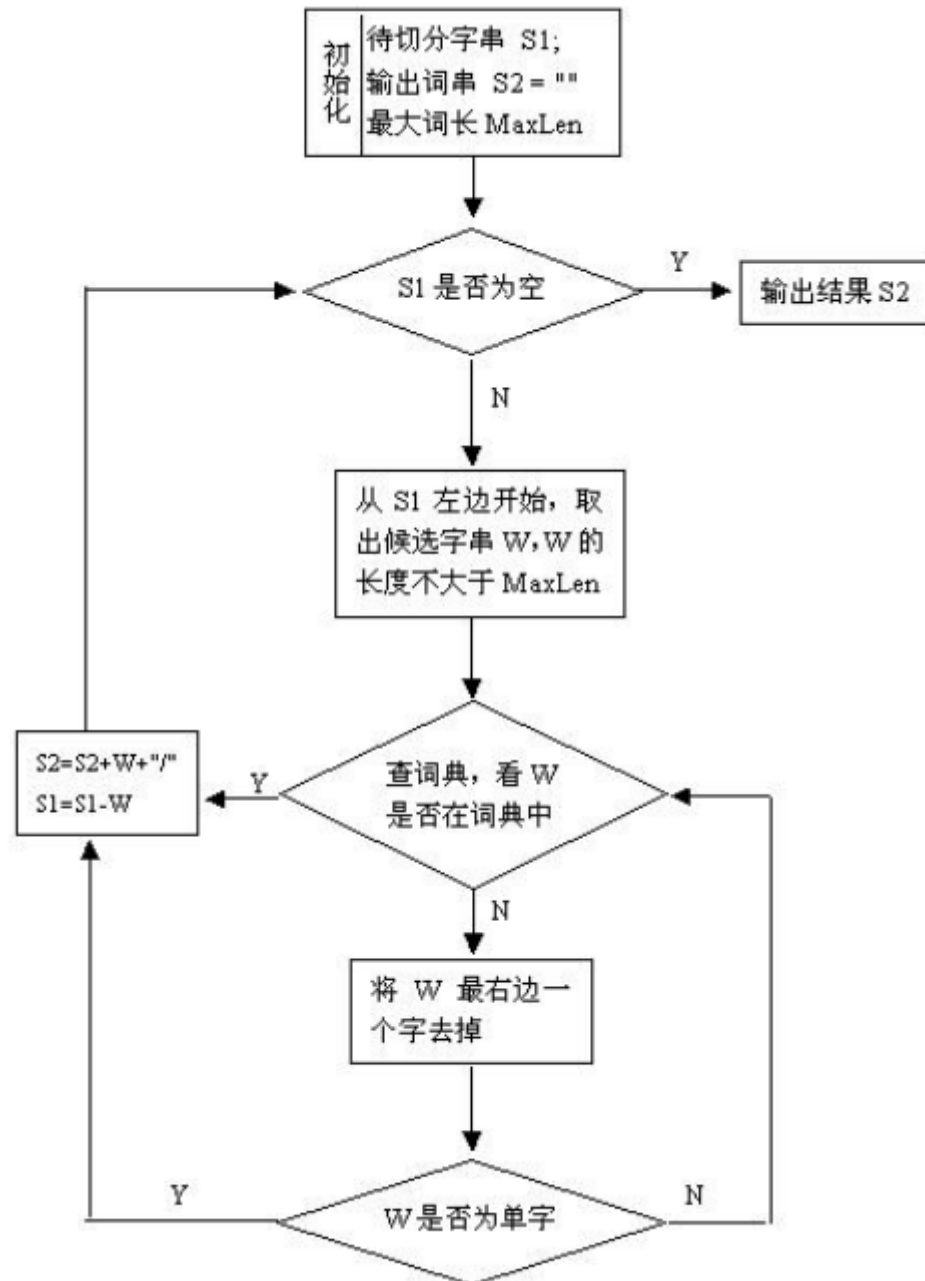


一. 正向最大匹配算法

1. 去除标点

2. 最大匹配

算法实现逻辑



```

1  # 最大正向匹配
2  def max_left_match(line, dict):
3      input_str = line
4      output_str = ""
5      # 最大词长
6      max_length = dict['max_length']
7      word_dict = dict['word_dict']
8      while input_str.strip() != '':
9          num = max_length
10         w = input_str[0:num]
11         while w not in word_dict:
12             num -= 1
13             w = w[0:num]
14             if len(w) == 1:
15                 break

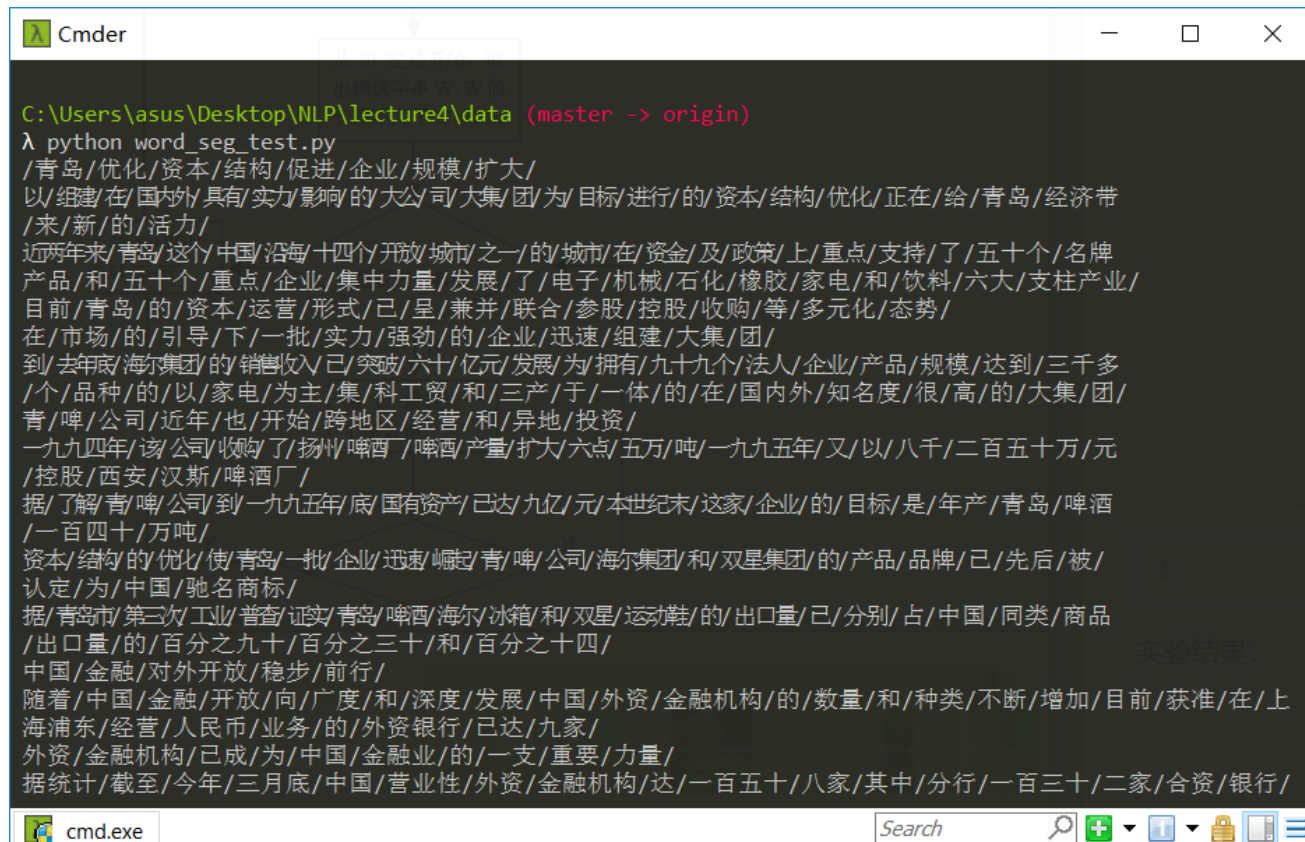
```

```

16         output_str += w + "/"
17         input_str = input_str[len(w):]
18     return output_str

```

实验结果：



```

C:\Users\asus\Desktop\NLP\lecture4\data (master -> origin)
λ python word_seg_test.py
/青岛/优化/资本/结构/促进/企业/规模/扩大/
以/组建/在/国内外/具有/实力/影响/的/大/公/司/大/集/团/为/目标/进行/的/资本/结构/优化/正在/给/青岛/经济带
/来/新/的/活力/
近两年来/青岛/这个/中国/沿海/十四个/开放/城市/之一/的/城市/在/资金/及/政策/上/重点/支持/了/五十个/名牌
产品/和/五十个/重点/企业/集中/力量/发展/了/电子/机械/石化/橡胶/家电/和/饮料/六大/支柱产业/
目前/青岛/的/资本/运营/形式/已/呈/兼并/联合/参股/控股/收购/等/多元化/态势/
在/市场/的/引导/下/一批/实力/强劲/的/企业/迅速/组建/大/集/团/
到/去年/底/海/尔/集/团/的/销售/收/入/已/突/破/六/十/亿/元/发/展/为/拥/有/九/十/九/个/法/人/企/业/产/品/规/模/达/到/三/千/多
/个/品/种/的/以/家/电/为/主/集/科/工/贸/和/三/产/于/一/体/的/在/国/内/外/知/名/度/很/高/的/大/集/团/
青/啤/公/司/近/年/也/开/始/跨/地/区/经/营/和/异/地/投/资/
一/九/九/四/年/该/公/司/收/购/了/扬/州/啤/酒/啤/酒/产/量/扩/大/六/点/五/万/吨/一/九/九/五/年/又/以/八/千/二/百/五/十/万/元
/控/股/西/安/汉/斯/啤/酒/厂/
据/了/解/青/啤/公/司/到/一/九/九/五/年/底/国/有/资/产/已/达/九/亿/元/本/世/纪/末/这/家/企/业/的/目/标/是/年/产/青/岛/啤/酒
/一/百/四/十/万/吨/
资/本/结/构/的/优/化/使/青/岛/一/批/企/业/迅/速/崛/起/青/啤/公/司/海/尔/集/团/和/双/星/集/团/的/产/品/品/牌/已/先/后/被/
认/定/为/中/国/驰/名/商/标/
据/青/岛/市/第/三/次/工/业/普/查/证/实/青/岛/啤/酒/海/尔/冰/箱/和/双/星/运/动/鞋/的/出/口/量/已/分/别/占/中/国/同/类/商/品
/出/口/量/的/百/分/之/九/十/百/分/之/三/十/和/百/分/之/十/四/
中/国/金/融/对/外/开/放/稳/步/前/行/
随/着/中/国/金/融/开/放/向/广/度/和/深/度/发/展/中/国/外/资/金/融/机/构/的/数/量/和/种/类/不/断/增/加/目/前/获/准/在/上
海/浦/东/经/营/人/民/币/业/务/的/外/资/银/行/已/达/九/家/
外/资/金/融/机/构/已/成/为/中/国/金/融/业/的/一/支/重/要/力/量/
据/统/计/截/至/今/年/三/月/底/中/国/营/业/性/外/资/金/融/机/构/达/一/百/五/十/八/家/其/中/分/行/一/百/三/十/二/家/合/资/银/行/

```

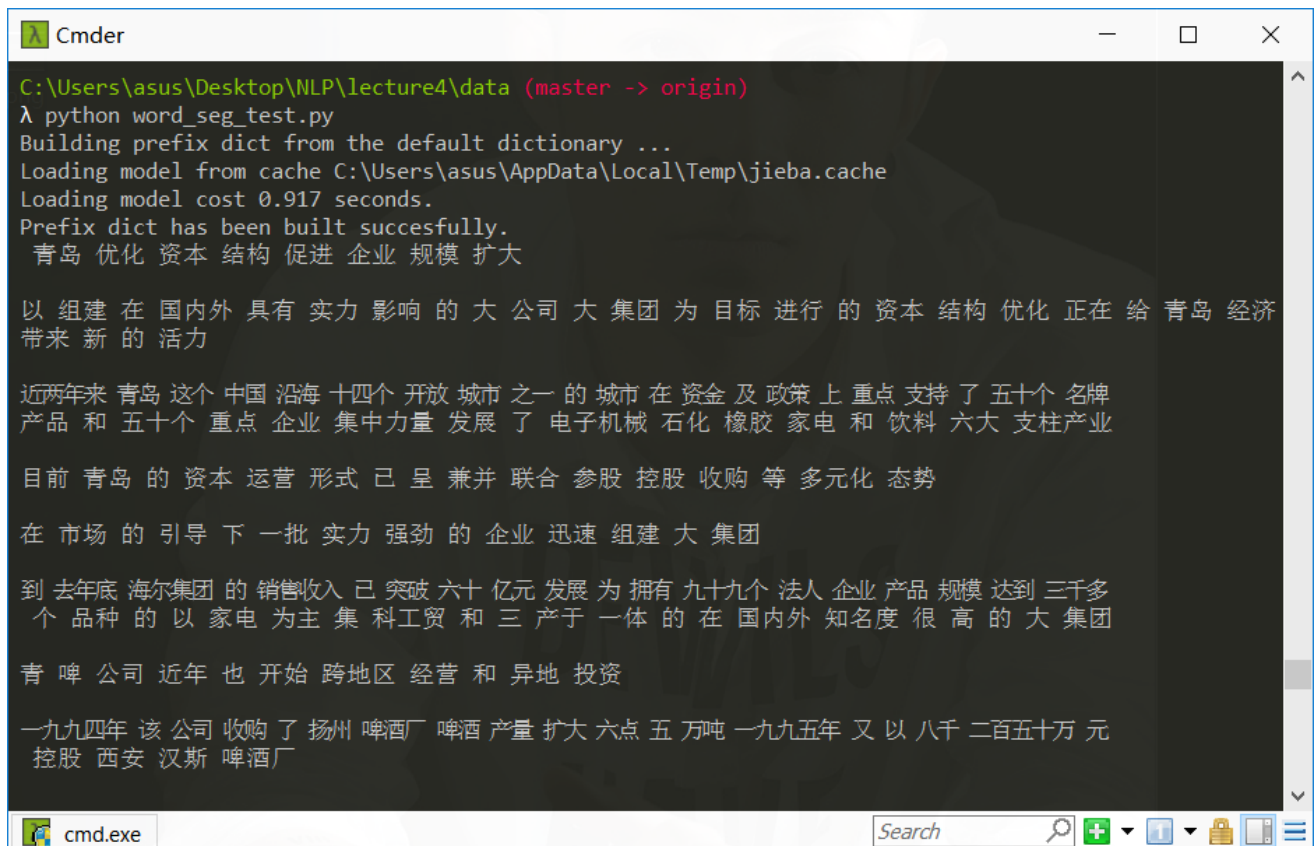
3.利用jieba库的分词功能

```

1  # 利用jieba库的分词功能
2  def jieba_cut(line):
3      line_seg = " ".join(jieba.cut(line))
4      return line_seg

```

实验结果：



```
C:\Users\asus\Desktop\NLP\lecture4\data (master -> origin)
λ python word_seg_test.py
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\asus\AppData\Local\Temp\jieba.cache
Loading model cost 0.917 seconds.
Prefix dict has been built succesfully.
青岛 优化 资本 结构 促进 企业 规模 扩大

以 组建 在 国内外 具有 实力 影响 的 大 公司 大 集团 为 目标 进行 的 资本 结构 优化 正在 给 青岛 经济
带来 新 的 活力

近两年来 青岛 这个 中国 沿海 十四个 开放 城市 之一 的 城市 在 资金 及 政策 上 重点 支持 了 五十个 名牌
产品 和 五十个 重点 企业 集中力量 发展 了 电子机械 石化 橡胶 家电 和 饮料 六大 支柱产业

目前 青岛 的 资本 运营 形式 已 呈 兼并 联合 参股 控股 收购 等 多元化 态势

在 市场 的 引导 下 一批 实力 强劲 的 企业 迅速 组建 大 集团

到 去年底 海尔集团 的 销售收入 已 突破 六十 亿元 发展 为 拥有 九十九个 法人 企业 产品 规模 达到 三千多
个 品种 的 以 家电 为主 集 科工贸 和 三 产于 一体 的 在 国内外 知名度 很 高 的 大 集团

青 啤 公司 近年 也 开始 跨地区 经营 和 异地 投资

一九九四年 该 公司 收购 了 扬州 啤酒厂 啤酒 产量 扩大 六点 五 万吨 一九九五年 又 以 八千 二百五十万 元
控股 西安 汉斯 啤酒厂
```

4.完整代码

```
1 # encoding=utf-8
2 import nltk
3 import string
4 import re
5 import jieba
6
7 # 加载字典
8 def load_word_list():
9     max_length = 0
10    word_dict = set()
11    for line in open('./data/corpus.dict.txt',encoding='utf-
12    8',errors='ignore').readlines():
13        tmp = len(line)
14        if(max_length < tmp):
15            max_length = tmp
16        word_dict.add(line.strip())
17    return {
18        'max_length':max_length,
19        'word_dict':word_dict
20    }
21
22 # 去标点
23 def filter_punctuation(line):
24     # 去除标点符号
```

```

24 punc = "[! ? . , \" # $ % & ' ( ) * + , - / : ; < = > @ [ \ ] ^ _ ` { | } ~ 《 》 「 」 『 』 【 】 〈 〉 〔 〕 √ ° „ ∼ ☞ ☛ -- ‘ ’ “ ” „ … … ! \ " # $ % & \ ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~ ] + "
25     line = re.sub(punc, "",line)
26     return line
27
28 # 最大正向匹配
29 def max_left_match(line, dict):
30     input_str = line
31     output_str = ""
32     # 最大词长
33     max_length = dict['max_length']
34     word_dict = dict['word_dict']
35     while input_str.strip() != '':
36         num = max_length
37         w = input_str[0:num]
38         while w not in word_dict:
39             num -= 1
40             w = w[0:num]
41             if len(w) == 1:
42                 break
43         output_str += w + "/"
44         input_str = input_str[len(w):]
45     return output_str
46
47 # 利用jieba库的分词功能
48 def jieba_cut(line):
49     line_seg = " ".join(jieba.cut(line))
50     return line_seg
51
52 # 测试
53 def main():
54     dict = load_word_list()
55     for line in open('./data/corpus.sentence.txt',encoding='utf-
56     8',errors='ignore').readlines():
57         # 去标点
58         new_line = filter_punctuation(line)
59         # 自己写的最大匹配
60         result = max_left_match(new_line, dict)
61         # jieba库的分词
62         #result = jieba_cut(new_line)
63         # 结果
64         print(result)
65
66 if __name__ == '__main__':
67     main()
68     # print(__name__)

```