

华北水利水电大学

自然语言处理

实验报告

2022——2023 学年

第一 学期

实验报告序号： 实验报告（二）

实 验 名 称： 文本信息抽取的实现

学生专业班级： 人工智能 2020185

学 生 姓 名： 高树林

学 号： 202018526

专 业： 人 工 智 能

一、实验目的

- 1.掌握依存句法分析的实现方法。
- 2.能够熟练地运用句法分析技术，对文本中的特定关系类型进行抽取。

二、实验内容

给定待抽取商品标签的热水器评论数据 `comment.csv`（其中“评论”列保存了用户对该品牌热水器的评论），采用句法分析技术抽取各条评论数据的关键标签（如价格便宜、质量不错等），并进行统计，最后输出出现频率最高的 20 种标签。

三、实验要求

- (1)数据选取要求：从数据集中随机选取 2000 篇文档。
- (2)实验报告撰写：以分析为主，写问题分析和句法分析规则设计的思考，画流程图，不要放太多代码，关键代码或流程应有文字说明或解释。

四、问题分析

4.1 采用句法分析技术进行关系抽取的实现流程分析

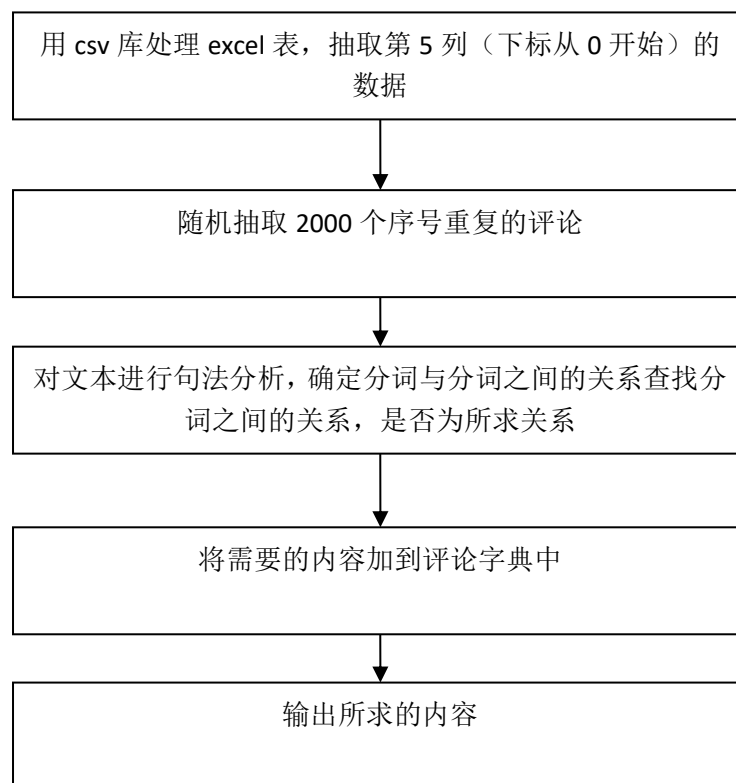


图 1 文本信息抽取流程图

4.2 对抽取到的关系进行统计的实现流程分析

流程 1： 由于 csv 文件不能直接提取其中的内容，需要将其转化成可以被 python 操作的数据类型，一般将其转为列表类型。

流程 2：抽取 2000 条评论内容可以先生成 2000 个不重复的 0 到 csv 表长度的数据，之后用列表生成式获取下标为这 2000 个数的评论列表数据。

流程 3：对文本进行句法分析，确定分词与分词之间的关系。结合所需求的分词关系，将对应的文本抽取出来。

流程 4：将流程 3 中抽取出来的文本放到字典里。以评论为键，出现次数为值，这样就能将评论和其对应的频率一起统计起来了。之后将字典按照值的大小进行降序排序。取前 20 的排序的结果基为最终所求的出现频率最高的 20 种标签。

五、关键代码

流程 1：用 csv 库处理 csv 表格，使用 `csv.read()` 函数读取 csv 文件，返回的是 `_io.TextIOWrapper` 类型的变量，这个变量用 `csv.reader()` 函数读取成为 `csv.reader` 类型的变量，这个变量相当与整个 csv 表格的内容。之后用 for 循环对该变量进行遍历，选取第 5 列（因为第 5 列是评论列，下标从 0 开始）的内容。新设置一个评论列表 `comments`，将该内容用列表的 `append()` 方法添加到 `comments` 列表里面，就能得到只含有评论的列表，每一个元素是一个评论。

流程 2：抽取 2000 个序号不同的评论，即在流程 1 中的评论元素列表 `comments` 里面选取 2000 个元素。新建一个新的空列表，用来装 2000 个评论，其元素可以通过下标引用，于是可以将其放入 while 循环里面。循环结束的条件为新列表中有 2000 条评论，即新列表的长度为 2000。用 `random` 库生成随机数，每一次生成一个在 0 到 `len(comments)` 的数，如果这个数未生成过，将它加到列表里面，如果生成过，就重新生成随机数。知道最后生成 2000 个数字。2000 个数字生成后，使用列表生成式就能获取 2000 条评论，将该评论列表作为当前函数返回的结果。

```
1. def GetDate(path,num):
2.     with open(path, "r", encoding='UTF-8') as file:
3.         list1 = []
4.         for row in csv.reader(file):
5.             list1.append(row[5])
6.         del list1[0]
7.         a = []
8.         while len(a) != num:
```

```

9.     b = random.randint(0, len(list1))
10.    if b not in a:
11.        a.append(b)
12.    sorted(a)
13.    return [list1[i] for i in a]

```

流程 3：先规定分词之间的关系，至少规定在分词时需要使用到的关系，如主谓关系、定中关系、动宾关系等等，关系定义好之后就能对流程 2 里的包含 2000 条评论的列表的元素进行分析，利用 `ltp` 库中的 `seg()` 方法实现句子分词，返回的两个对象中，`hidden` 对象被 `dep()` 方法作用得到句法分析的每个词之间的关系。需要注意的是在依存句法当中，虚节点 `ROOT` 占据了 0 位置，因此节点的下标从 1 开始。之后比较两个词之间是否具有需要抽取的关系（如主谓关系、状中关系、定中关系）如果满足，就说明找到了评论中需要抽取的文本对象，将其抽取出来。

```

1. for sentence in GetDate('comment.csv',2000):
2.     seg, hidden = ltp.seg([sentence])
3.     dep = ltp.dep(hidden)
4.     sent=seg[0]
5.     dep0=dep[0]

```

上述代码的作用是获取每个词之间的关系，其中 `seg`、`dep` 返回的是一个二维列表，因此 4、5 两行需要用到取第一个元素的操作，将二维列表转化成一维列表，因为后续的操作对象不能为二维列表。

```

1. for w1,w2,r in dep0:
2.     if r=='SBV':
3.         if (w2-1,w2,'ADV') in dep0:
4.             if ( w1-1,w1,'ATT') in dep0:
5.                 comment =sent[w1-1-1] +sent[w1-1]+sent[w2-1-1]+sent[w2-1]
6.                 comments[comment] = 1 if comment not in comments else comments[comment] + 1
7.             else:
8.                 comment = sent[w1-1]+sent[w2-1-1]+sent[w2-1]
9.                 comments[comment] = 1 if comment not in comments else comments[comment] + 1
10.    if r=='CMP':
11.        comment = sent[w2-1]+sent[w1-1]
12.        comments[comment] = 1 if comment not in comments else comments[comment] + 1

```

流程 4：对流程 3 得到的字典按值进行排序，其中值代表了抽取结果为改键的次数，按照这个顺序排列的字典，只需要输出前 20 个元素的键就是所求的评论中出现频率最高的 20 种标签。

```

1. d_order=sorted(comments.items(),key=lambda x:x[1],reverse=True)
2. for i in range(20):

```

3. `print(d_order[i][0])`

六. 实验结果与分析

实验结果:

利用句法分析对文本进行抽取的结果如下图 2 所示。

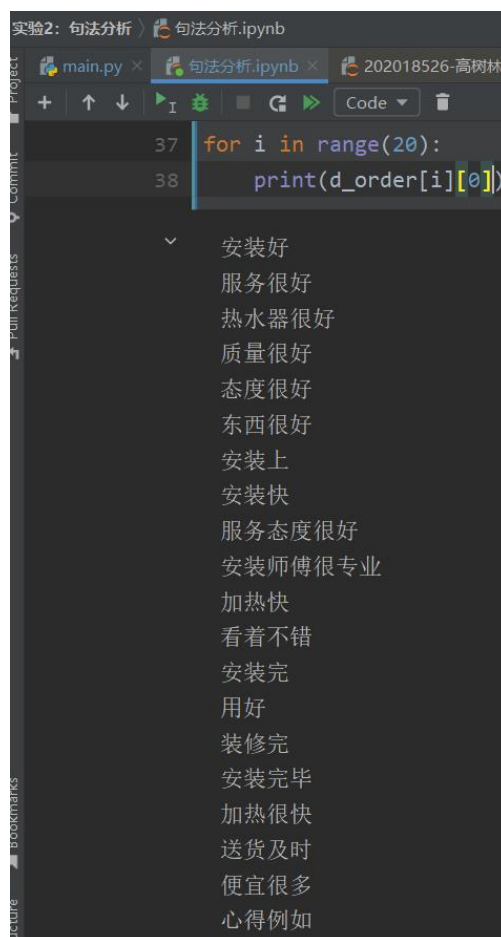
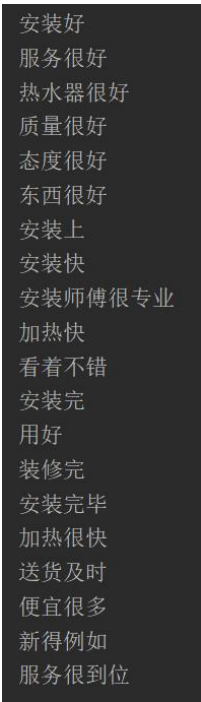


图 2 句法分析抽取文本关键词

对句法分析抽取文本的评价如下：从流程上来看，利用句法分析抽取文本获得标签的方法操作简单，容易实现。但是从结果上看，他并没有得到一个很惊艳的结果，对于某些意思相近的词语它并不能区分得很好，比如上图 2 的第 2 行“服务很好”、第 5 行“服务态度很好”都可以被第 9 行“服务态度很好”标签包含掉，但是在句法分析中，这 3 个标签的主语分别是“服务”、“态度”、“服务态度”，这三个在实例中并不是完全一样的，此外，由于句法分析只考虑词性和它们之间的关系，并不会考虑最后抽取出来的内容是否符合正常逻辑，因此会出现如图 2 中“安装上”、“安装完”、“用好”、“装修完”、“安装完毕”、“心得例如”等等不明所以的抽取结果。当然，倘若加更多的条件语句，针对更对的关系进行处理和讨论，在理论上是可以减少这种结果的发生，但是这就完全脱离了自然语言处理的范畴了，因为自然语言处理一般是训练出一个含有参数的模型，这个模

型可以针对给出的任意句子做运算最终返回出正确的结果，至少为预期的结果。此外，句法分析用于抽取文本关键标签是需要很高的成本的，因为用句法分析抽取文本需要考虑到各个成分之间的关系，这需要操作者具有较强的语言理论基础，否则抽取出来的句子和标签是有问题的。总体来说，通过句法结构分析，我们就能够分析出语句的主干，以及各成分间关系。对于复杂语句，仅仅通过词性分析，不能得到正确的语句成分关系。

可能存在的提高标签抽取准确率的方法和手段：目前从算法角度来提高句法分析获取文本标签的研究处于全球的瓶颈，但是对于上述的一种标签包含另一种标签的现象，一种解决方法是将抽取结果按照主语分类，把主语一样的内容全部放在一个以主语为键的字典里，比较字典里的键是否具有相互包含关系，如果存在包含关系，就将长键值并入到短键值里，这样就能解决评论中标签主语相互包含的现象了。改进后输出结果如下图 3 所示，可以明显看到，图 2 种的“服务态度很好”标签已经加入到“服务很好”和“态度很好”中了。



安装好
服务很好
热水器很好
质量很好
态度很好
东西很好
安装上
安装快
安装师傅很专业
加热快
看着不错
安装完
用好
装修完
安装完毕
加热很快
送货及时
便宜很多
新得例如
服务很到位

图 3 改进后输出结果