L6-Chinese-Text-Processing

2018 年 4 月 1 日

1 中文分词 (Chinese Word Segmentation)

中文分词 (Chinese Word Segmentation) 指的是将一个汉字序列切分成一个一个单独的词。分词就是将连续的字序列按照一定的规范重新组合成词序列的过程。我们知道，在英文的行文中，单词之间是以空格作为自然分界符的，而中文只是字、句和段能通过明显的分界符来简单划界，唯独词没有一个形式上的分界符，虽然英文也同样存在短语的划分问题，不过在词这一层上，中文比之英文要复杂得多、困难得多。

- 中文分词与英文分词

与英文为代表的拉丁语系语言相比，英文以空格作为天然的分隔符，而中文由于继承自古代汉语的传统，词语之间没有分隔。 古代汉语中除了连绵词和人名地名等，词通常就是单个汉字，所以当时没有分词书写的必要。而现代汉语中双字或多字词居多，一个字不再等同于一个词。

- 在中文里，“词” 和 “词组” 边界模糊

现代汉语的基本表达单元虽然为 “词”，且以双字或者多字词居多，但由于人们认识水平的不同，对词和短语的边界很难去区分。例如：“对随地吐痰者给予处罚”，“随地吐痰者” 本身是一个词还是一个短语，不同的人会有不同的标准，同样的 “海上”“酒厂” 等等，即使是同一个人也可能做出不同判断，如果汉语真的要分词书写，必然会出现混乱，难度很大。

- 中文分词的应用

中文分词的方法其实不局限于中文应用，也被应用到英文处理，如手写识别，单词之间的空格就不很清楚，中文分词方法可以帮助判别英文单词的边界。

1.1 中文分词算法分类

现有的分词算法可分为三大类：

- 基于字符串匹配的分词方法

- 基于理解的分词方法
- 基于统计的分词方法。

按照是否与词性标注过程相结合，又可以分为**单纯分词方法**和**分词与标注相结合**的一体化方法。

到底哪种分词算法的准确度更高，目前并无定论。对于任何一个成熟的分词系统来说，不可能单独依靠某一种算法来实现，都需要综合不同的算法。例如，海量科技的分词算法就采用“复方分词法”，所谓复方，就是像中西医结合般综合运用机械方法和知识方法。对于成熟的中文分词系统，需要多种算法综合处理问题。

有了成熟的分词算法，是否就能容易的解决中文分词的问题呢？事实远非如此。中文是一种十分复杂的语言，让计算机理解中文语言更是困难。在中文分词过程中，有两大难题一直没有完全突破。

- 歧义识别

歧义是指同样的一句话，可能有两种或者更多的切分方法。主要的歧义有两种：交集型歧义和组合型歧义，例如：表面的，因为“表面”和“面的”都是词，那么这个短语就可以分成“表面的”和“表面的”。这种称为交集型歧义（交叉歧义）。像这种交集型歧义十分常见，前面举的“和服”的例子，其实就是因为交集型歧义引起的错误。“化妆和服装”可以分成“化妆和服装”或者“化妆和服装”。由于没有人的知识去理解，计算机很难知道到底哪个方案正确。

- 新词识别

命名实体（人名、地名）、新词，专业术语称为未登录词。也就是那些在分词词典中没有收录，但又确实能称为词的那些词。最典型的是人名，人可以很容易理解。句子“王军虎去广州了”中，“王军虎”是个词，因为是一个人的名字，但要是让计算机去识别就困难了。如果把“王军虎”做为一个词收录到字典中去，全世界有那么多名字，而且每时每刻都有新增的人名，收录这些人名本身就是一项既不划算又巨大的工程。即使这项工作可以完成，还是会存在问题，例如：在句子“王军虎头虎脑的”中，“王军虎”还能不能算词？

除了人名以外，还有机构名、地名、产品名、商标名、简称、省略语等都是很难处理的问题，而且这些又正好是人们经常使用的词，因此对于搜索引擎来说，分词系统中的新词识别十分重要。新词识别准确率已经成为评价一个分词系统好坏的重要标志之一。

1.2 中文分词工具示例

1.2.1 “结巴”中文分词 (jieba)

- 支持三种分词模式：
  - 精确模式，试图将句子最精确地切开，适合文本分析；

  - 全模式，把句子中所有的可以成词的词语都扫描出来, 速度非常快，但是不能解决歧义；
  - 搜索引擎模式，在精确模式的基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词。

- 支持繁体分词
- 支持自定义词典

“结巴” 中文分词算法

- 基于前缀词典实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图 (DAG)
- 采用了动态规划查找最大概率路径, 找出基于词频的最大切分组合
- 对于未登录词，采用了基于汉字成词能力的 HMM 模型，使用了 Viterbi 算法

1.2.2 Jieba 分词基本模式

jieba.cut 方法接受三个输入参数: 需要分词的字符串；cut_all 参数用来控制是否采用全模式；HMM 参数用来控制是否使用 Hidden Markov Model (HMM) 模型

```
In [7]: import jieba

        # 全模式
        seg_list = jieba.cut("我来到北京清华大学", cut_all=True)
        print("Full Mode: " + "/ ".join(seg_list))
```

```
Full Mode: 我/ 来到/ 北京/ 清华/ 清华大学/ 华大/ 大学
```

```
In [6]: # 精确模式
        seg_list = jieba.cut("我来到北京清华大学", cut_all=False)
        print("Default Mode: " + "/ ".join(seg_list))
```

```
Default Mode: 我/ 来到/ 北京/ 清华大学
```

```
In [5]: # 搜索引擎模式
        seg_list = jieba.cut_for_search("小明硕士毕业于中国科学院计算所，后在日本京都大学深造")
        print(", ".join(seg_list))
```

```
小明，硕士，毕业，于，中国，科学，学院，科学院，中国科学院，计算，计算所，，，后，在，日本，京都，
```

1.2.3 添加自定义词典

开发者可以指定自己自定义的词典，以便包含 jieba 词库里没有的词。虽然 jieba 有新词识别能力，但是自行添加新词可以保证更高的正确率。词典格式为一个词占一行；每一行分三部分：词语、词频（可省略）、词性（可省略），用空格隔开，顺序不可颠倒。

在使用 jieba 分词时经常会发现一些未登录词，因此增加领域词表就变得很重要，下面提供增加几种途径：

- 所在领域权威词汇字典
- 搜狗细胞词库、百度输入法领域词库
- 手动添加字典

```
In [8]: test_sent = (
        "李小福是创新办主任也是云计算方面的专家；什么是八一双鹿\n"
        "例如我输入一个带“韩玉赏鉴”的标题，在自定义词库中也增加了此词为 N 类\n"
        "「台中」正確應該不會被切開。mac 上可分出「石墨烯」；此時又可以分出來凱特琳了。"
        )
```

```
In [9]: # 未使用个人字典
        words = jieba.cut(test_sent)
        print('/'.join(words))
```

```
李小福/是/创新/办/主任/也/是/云/计算/方面/的/专家/;/ /什么/是/八/一双/鹿/
/例如/我/输入/一个/带/“/韩玉/赏鉴/”/的/标题/，/在/自定义词/库中/也/增加/了/此/词为/N/类/
/「/台/中/」/正確/應該/不會/被/切開/。/mac/上/可/分出/「/石墨/烯/」/；/此時/又/可以/分出/來凱/特琳
```

```
In [14]: # 加载个人字典后的结果
         jieba.load_userdict("userdict.txt")
         words = jieba.cut(test_sent)
         print('/'.join(words))
```

```
李小福/是/创新办/主任/也/是/云计算/方面/的/专家/;/ /什么/是/八一双鹿/
/例如/我/输入/一个/带/“/韩玉赏鉴/”/的/标题/，/在/自定义词/库中/也/增加/了/此/词为/N/类/
/「/台中/」/正確/應該/不會/被/切開/。/mac/上/可/分出/「/石墨/烯/」/；/此時/又/可以/分出/來/凱特琳/
```

2 关键词提取 (Key Word Extraction)

关键词提取说白了就是对文章进行总结，从一篇文章中抽取出比较重要的一些词汇，帮助阅读者高效率地了解文章的大意。尤其是对互联网环境下，每天大量的信息涌出，若不加以预处理，则会成为网页浏览者的负担。关键词提取技术可以非常简单，也可以非常复杂，但是其任务框架都一样，输入一个文章，输出几个关键词。那么给定一篇文章，关键词是怎么抽取出来的呢？

当前关键词提取算法主要可以分成两个流派：

- Statistical Based

其思路是，先定义一个关键词指标，然后为文章中所有词汇计算关键词指标，把词汇按照指标从大到小排列，指标大的优先选为关键词。这种思路很简单，有点儿像班里选班长，老师说选学习成绩最好的同学当班长，那么选拔方法就是，先定义一个指标，比如考试总分，然后，给每个学生的语文、数学、物理成绩加总，排名，总分排第一的当班长（关键词）。当然，老师可以指定班里有任意多个班长，例如选 K 个班长，那么就是成绩排名前 K 个学生当班长。统计流派的关键在于，计算每个词汇的关键词指标，这个指标是根据词汇在文章中的表现统计并计算出来的，所以有 Statistical 其名。

在对关键词进行提取时，可以有多种指标，影响力最大的两个是：TF-IDF 指标和 PageRank 指标。TF-IDF 基于词袋模型（Bag-of-Words），把文章表示成词汇的集合，由于集合中词汇元素之间的顺序位置与集合内容无关，所以 TF-IDF 指标不能有效反映文章内部的词汇组织结构。PageRank 指标，基于网络模型（Graph Model），把文章表示成网络的结构，网络中的节点表示词汇，节点之间的边为词汇之间的位置邻接关系，网络结构比集合结构包含信息多，考虑了文章内词汇的顺序，所以 PageRank 指标一般比 TF-IDF 指标表现更好。

- Rule Based

其思路是，将关键词提取任务，定义为一个对词汇进行**二分类**的任务。即给定一个词汇，要么是关键词，要么不是关键词，对其分类，是关键词为 1，不是就是 0。那么，关键词提取问题就变成了一个预测问题。预测问题，需要一个预测函数，这个函数就是规则，给定词汇，获得该词汇的特征，然后预测该词汇是否为关键词。

规则可以人工指定，也可以通过机器学习 (Machine Learning) 的方法获得。人工指定规则，一般比较难，费脑子，谁也不清楚究竟到底啥样儿的词是关键词。所以大家就想着让程序自己去获得规则，即通过机器学习。机器学习的方法相对省脑子，但是费体力，要手工标关键词，然后把标记过的样本放到模型里去把规则给学习出来。机器学习过程中，需要指定一些词汇特征，用于训练。这些特征，一般也不知道，所以也需要人工指定，比如，考虑词频、词汇包含的字数、词性、词汇的位置等等。现在深度学习成为关键词提取新的发展方向，但是深度学习的方法只能通过复杂网络的训练帮你抽象出词汇特征，但是还得依赖于人工标注，依然需要人去标记文章。

2.1 Jieba 基于 TF-IDF 算法的关键词抽取示例

```
In [31]: import jieba.analyse
         sentence = (
         "苍茫的天涯是我的爱，绵绵的青山脚下花正开，什么样的节奏是最呀最摇摆，\
         什么样的歌声才是最开怀，弯弯的河水从天上来，流向那万紫千红一片海，\
         哗啦啦的歌谣是我们的期待，一路边走边唱才是最自在，我们要唱就要唱得最痛快，\
         你是我天边最美的云彩，让我用心把你留下来，悠悠的唱着最炫的民族风，\
         让爱卷走所有的尘埃，你是我心中最美的云彩，怎么没就让你留下来，\
         永远都唱着最炫的民族风，是整片天空最美的姿态，我听见你心中永远的天籁，\
         登上天外云霄的舞台")

         jieba.analyse.extract_tags(sentence, topK=15, withWeight=False, allowPOS=())

Out[31]: [' 最美',
          ' 云彩',
          ' 留下来',
          ' 什么样',
          ' 永远',
          ' 花正开',
          ' 爱卷',
          ' 悠悠的',
          ' 万紫千红',
          ' 民族',
          ' 整片',
          ' 天外',
          ' 心中',
          ' 开怀',
          ' 弯弯的']
```

2.2 Jieba 基于 TextRank 算法的关键词抽取

其基本思想是将待抽取关键词的文本进行分词，以固定窗口大小 (默认为 5，通过 span 属性调整)，词之间的共现关系，构建图计算图中节点的 PageRank，注意是无向带权图。

```
In [32]: jieba.analyse.textrank(sentence, topK=15, withWeight=False,
                                 allowPOS=('ns', 'n', 'vn', 'v'))

Out[32]: [' 留下来',
          ' 登上',
```

' 整片',
' 姿态',
' 歌声',
' 开怀',
' 歌谣',
' 期待',
' 天空',
' 舞台',
' 用心',
' 就让',
' 天籁',
' 河水',
' 摇摆']