

**计 控 学 院**

**College of computer and control engineering** **Qiqihar university**

基于jieba分词生成词云分析及应用

实训报告

**专业班级** 计本163

**学生姓名** 张清

**学 号** 2016021214

**指导教师** 李丽丽

**提交日期**

**成 绩**

目录

[基于jieba分词的小说关键词生成词云](#_Toc19615_WPSOffice_Level1) [1](#_Toc19615_WPSOffice_Level1)

[1. 概述](#_Toc14068_WPSOffice_Level1) [2](#_Toc14068_WPSOffice_Level1)

[1.1 目的](#_Toc14068_WPSOffice_Level2) [2](#_Toc14068_WPSOffice_Level2)

[1.2 背景](#_Toc17985_WPSOffice_Level2) [2](#_Toc17985_WPSOffice_Level2)

[1.3 研究内容](#_Toc636_WPSOffice_Level2) [3](#_Toc636_WPSOffice_Level2)

[2 环境说明](#_Toc17985_WPSOffice_Level1) [3](#_Toc17985_WPSOffice_Level1)

[2.1 环境要求](#_Toc23382_WPSOffice_Level2) [3](#_Toc23382_WPSOffice_Level2)

[2.2 安装与搭建环境](#_Toc1115_WPSOffice_Level2) [3](#_Toc1115_WPSOffice_Level2)

[3．需求分析](#_Toc636_WPSOffice_Level1) [3](#_Toc636_WPSOffice_Level1)

[3.1找到目标小说](#_Toc6655_WPSOffice_Level2) [4](#_Toc6655_WPSOffice_Level2)

[3.2对小说进行分词统计](#_Toc25769_WPSOffice_Level2) [4](#_Toc25769_WPSOffice_Level2)

[3.3对结果进行统计](#_Toc26843_WPSOffice_Level2) [4](#_Toc26843_WPSOffice_Level2)

[3.4生成词云和柱状图](#_Toc23266_WPSOffice_Level2) [4](#_Toc23266_WPSOffice_Level2)

[4．概要设计](#_Toc23382_WPSOffice_Level1) [4](#_Toc23382_WPSOffice_Level1)

[4.1为文本设置编码，](#_Toc18278_WPSOffice_Level2) [4](#_Toc18278_WPSOffice_Level2)

[4.2使用分词方法获取数据](#_Toc16062_WPSOffice_Level2) [4](#_Toc16062_WPSOffice_Level2)

[4.3使用函数生成词云](#_Toc21500_WPSOffice_Level2) [4](#_Toc21500_WPSOffice_Level2)

[5．详细设计](#_Toc1115_WPSOffice_Level1) [4](#_Toc1115_WPSOffice_Level1)

[5.1 数据抓取](#_Toc19619_WPSOffice_Level2) [4](#_Toc19619_WPSOffice_Level2)

[1. 引擎(EGINE)](#_Toc14068_WPSOffice_Level3) [4](#_Toc14068_WPSOffice_Level3)

[2.调度器(SCHEDULER)](#_Toc17985_WPSOffice_Level3) [5](#_Toc17985_WPSOffice_Level3)

[3. 下载器(DOWLOADER)](#_Toc636_WPSOffice_Level3) [5](#_Toc636_WPSOffice_Level3)

[4.爬虫(SPIDERS)](#_Toc23382_WPSOffice_Level3) [5](#_Toc23382_WPSOffice_Level3)

[5. 项目管道(ITEM PIPLINES)](#_Toc1115_WPSOffice_Level3) [5](#_Toc1115_WPSOffice_Level3)

[6. 下载器中间件(Downloader Middlewares)](#_Toc6655_WPSOffice_Level3) [5](#_Toc6655_WPSOffice_Level3)

[7. 爬虫中间件(Spider Middlewares)](#_Toc25769_WPSOffice_Level3) [5](#_Toc25769_WPSOffice_Level3)

[5.2 数据存储](#_Toc31480_WPSOffice_Level2) [5](#_Toc31480_WPSOffice_Level2)

[5.3 数据处理](#_Toc8904_WPSOffice_Level2) [6](#_Toc8904_WPSOffice_Level2)

[1.停用词处理:](#_Toc26843_WPSOffice_Level3) [6](#_Toc26843_WPSOffice_Level3)

[2.jieba分词:](#_Toc23266_WPSOffice_Level3) [6](#_Toc23266_WPSOffice_Level3)

[1. 支持 3 种分词模式：精确模式、全模式、搜索引擎模式](#_Toc18278_WPSOffice_Level3) [6](#_Toc18278_WPSOffice_Level3)

[2. 支持繁体分词](#_Toc16062_WPSOffice_Level3) [6](#_Toc16062_WPSOffice_Level3)

[3. 支持自定义词典](#_Toc21500_WPSOffice_Level3) [6](#_Toc21500_WPSOffice_Level3)

[5.4 数据可视化](#_Toc24542_WPSOffice_Level2) [7](#_Toc24542_WPSOffice_Level2)

[1.生成柱状图:](#_Toc19619_WPSOffice_Level3) [7](#_Toc19619_WPSOffice_Level3)

[2. 生成词云](#_Toc31480_WPSOffice_Level3) [7](#_Toc31480_WPSOffice_Level3)

[6．结论](#_Toc6655_WPSOffice_Level1) [8](#_Toc6655_WPSOffice_Level1)

[7．参考文献](#_Toc25769_WPSOffice_Level1) [8](#_Toc25769_WPSOffice_Level1)

基于jieba分词的小说关键词生成词云

摘要:从小说中利用jiebs分词统计出各词出现的频率，生成指定的词云。

关键字:结巴分词 词频统计 图表显示 词云生成

# 1. 概述

自然语言处理是[计算机](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA)科学领域与[人工智能](https://baike.baidu.com/item/%E4%BA%BA%E5%B7%A5%E6%99%BA%E8%83%BD/9180)领域中的一个重要方向。它研究能实现人与[计算机](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA/140338)之间用自然语言进行有效通信的各种理论和方法。自然语言处理是一门融[语言学](https://baike.baidu.com/item/%E8%AF%AD%E8%A8%80%E5%AD%A6/3632)、[计算机](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%A7%91%E5%AD%A6)[科学](https://baike.baidu.com/item/%E8%87%AA%E7%84%B6%E8%AF%AD%E8%A8%80%E5%A4%84%E7%90%86/365730?fr=aladdin)、[数学](https://baike.baidu.com/item/%E6%95%B0%E5%AD%A6/107037)于一体的[科学](https://baike.baidu.com/item/%E7%A7%91%E5%AD%A6)。因此，这一领域的研究将涉及[自然语言](https://baike.baidu.com/item/%E8%87%AA%E7%84%B6%E8%AF%AD%E8%A8%80/4146019)，即人们日常使用的[语言](https://baike.baidu.com/item/%E8%AF%AD%E8%A8%80/72744)，所以它与[语言学](https://baike.baidu.com/item/%E8%AF%AD%E8%A8%80%E5%AD%A6/3632)的研究有着密切的联系，但又有重要的区别。自然语言处理并不是一般地研究自然语言，而在于研制能有效地实现自然语言通信的[计算机系统](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%B3%BB%E7%BB%9F/7210959)，特别是其中的[软件系统](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E7%B3%BB%E7%BB%9F/224122)。因而它是计算机科学的一部分。

统计自然语言处理运用了[推测学](https://zh.wikipedia.org/w/index.php?title=%E6%8E%A8%E6%B8%AC%E5%AD%B8&action=edit&redlink=1" \o "推测学（页面不存在）)、[几率](https://zh.wikipedia.org/wiki/%E6%A9%9F%E7%8E%87" \o "几率)、[统计](https://zh.wikipedia.org/wiki/%E7%B5%B1%E8%A8%88" \o "统计)的方法来解决上述，尤其是针对容易高度模糊的长串句子，当套用实际文法进行分析产生出成千上万笔可能性时所引发之难题。处理这些高度模糊句子所采用消歧的方法通常运用到[语料库](https://zh.wikipedia.org/wiki/%E8%AA%9E%E6%96%99%E5%BA%AB" \o "语料库)以及[马可夫模型](https://zh.wikipedia.org/w/index.php?title=%E9%A6%AC%E5%8F%AF%E5%A4%AB%E6%A8%A1%E5%9E%8B&action=edit&redlink=1" \o "马可夫模型（页面不存在）)（Markov models）。统计自然语言处理的技术主要由同样自[人工智能](https://zh.wikipedia.org/wiki/%E4%BA%BA%E5%B7%A5%E6%99%BA%E6%85%A7" \o "人工智能)下与学习行为相关的子领域：[机器学习](https://zh.wikipedia.org/wiki/%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0" \o "机器学习)及[数据采掘](https://zh.wikipedia.org/wiki/%E8%B3%87%E6%96%99%E6%8E%A1%E6%8E%98" \o "数据采掘)所演进而成。

从网络下载或利用Python爬虫采集相应的小说，利用jieba分词包对词语进行分类，对小说词频做统计，生成词云图片。

## 1.1 目的

使学生能够学会使用jieba分词包，并统计词语生成的频率，将结果以词云形式展示。

1.2 背景

最早的自然语言理解方面的研究工作是机器翻译。1949年，美国人威弗首先提出了机器翻译设计方案。20世纪60年代，国外对机器翻译曾有大规模的研究工作，耗费了巨额费用，但人们当时显然是低估了自然语言的复杂性，语言处理的理论和技术均不成热，所以进展不大。主要的做法是存储两种语言的单词、短语对应译法的大辞典，翻译时一一对应，技术上只是调整语言的同条顺序。但日常生活中语言的翻译远不是如此简单，很多时候还要参考某句话前后的意思。

大约90年代开始，自然语言处理领域发生了巨大的变化。这种变化的两个明显的特征是：

（1）对系统输入，要求研制的自然语言处理系统能处理大规模的真实文本，而不是如以前的研究性系统那样，只能处理很少的词条和典型句子。只有这样，研制的系统才有真正的实用价值。

（2）对系统的输出，鉴于真实地理解自然语言是十分困难的，对系统并不要求能对自然语言文本进行深层的理解，但要能从中抽取有用的信息。例如，对自然语言文本进行自动地提取索引词，过滤，检索，自动提取重要信息，进行自动摘要等等。

同时，由于强调了“大规模”，强调了“真实文本”，下面两方面的基础性工作也得到了重视和加强。

（1）大规模真实语料库的研制。大规模的经过不同深度加工的真实文本的语料库，是研究自然语言统计性质的基础。没有它们，[统计方法](https://baike.baidu.com/item/%E7%BB%9F%E8%AE%A1%E6%96%B9%E6%B3%95)只能是无源之水。

（2）大规模、信息丰富的词典的编制工作。规模为几万，十几万，甚至几十万词，含有丰富的信息（如包含词的搭配信息）的计算机可用词典对自然语言处理的重要性是很明显的。

1.3 研究内容

**自然语言处理**（英语：**Natural Language Processing**，[缩写](https://zh.wikipedia.org/wiki/%E7%BC%A9%E5%86%99" \o "缩写)作 **NLP**）是[人工智能](https://zh.wikipedia.org/wiki/%E4%BA%BA%E5%B7%A5%E6%99%BA%E6%85%A7" \o "人工智能)和[语言学](https://zh.wikipedia.org/wiki/%E8%AA%9E%E8%A8%80%E5%AD%B8" \o "语言学)领域的分支学科。此领域探讨如何处理及运用[自然语言](https://zh.wikipedia.org/wiki/%E8%87%AA%E7%84%B6%E8%AA%9E%E8%A8%80" \o "自然语言)；自然语言处理包括多方面和步骤，基本有认知、理解、生成等部分。

自然语言认知和理解是让计算机把输入的[语言](https://zh.wikipedia.org/wiki/%E8%AA%9E%E8%A8%80" \o "语言)变成有意思的符号和关系，然后根据目的再处理。自然语言生成系统则是把计算机数据转化为自然语言。

由于理解（understanding）自然语言，需要关于外在世界的广泛知识以及运用操作这些知识的能力，自然语言认知，同时也被视为一个人工智能完备（AI-complete）的问题。同时，在自然语言处理中，"理解"的定义也变成一个主要的问题。

在自然语言处理过程中，为了能更好地处理句子，往往需要把句子拆开分成一个一个的词语，这样能更好的分析句子的特性，这个过程叫就叫做分词。由于中文句子不像英文那样天然自带分隔，并且存在各种各样的词组，从而使中文分词具有一定的难度。不过，中文分词并不追求完美，而是通过关键字识别技术，抽取句子中最关键的部分，从而达到理解句子的目的。

# 2 环境说明

2.1 环境要求

该实训具体在以下环境进行，Windows10 64位操作系统，12G内存，500G硬盘。

2.2 安装与搭建环境

硬件环境:Windows10,CPU(2.59GHZ),RAM(12G),ROM(500G)

软件环境: PyCharm2017 + Anaconda3.6.1

软件包需要使用：Numpy(处理图象)、pandas（统计词频）、jieba(分词)、matplotlib（绘图）、pylab（生成字体）、seaborn(生成柱状图)、PIL(处理图像)。

3．需求分析

3.1找到目标小说

3.2对小说进行分词统计

3.3对结果进行统计

3.4生成词云和柱状图

4．概要设计

4.1为文本设置编码，

4.2使用分词方法获取数据

4.3使用函数生成词云

5．详细设计

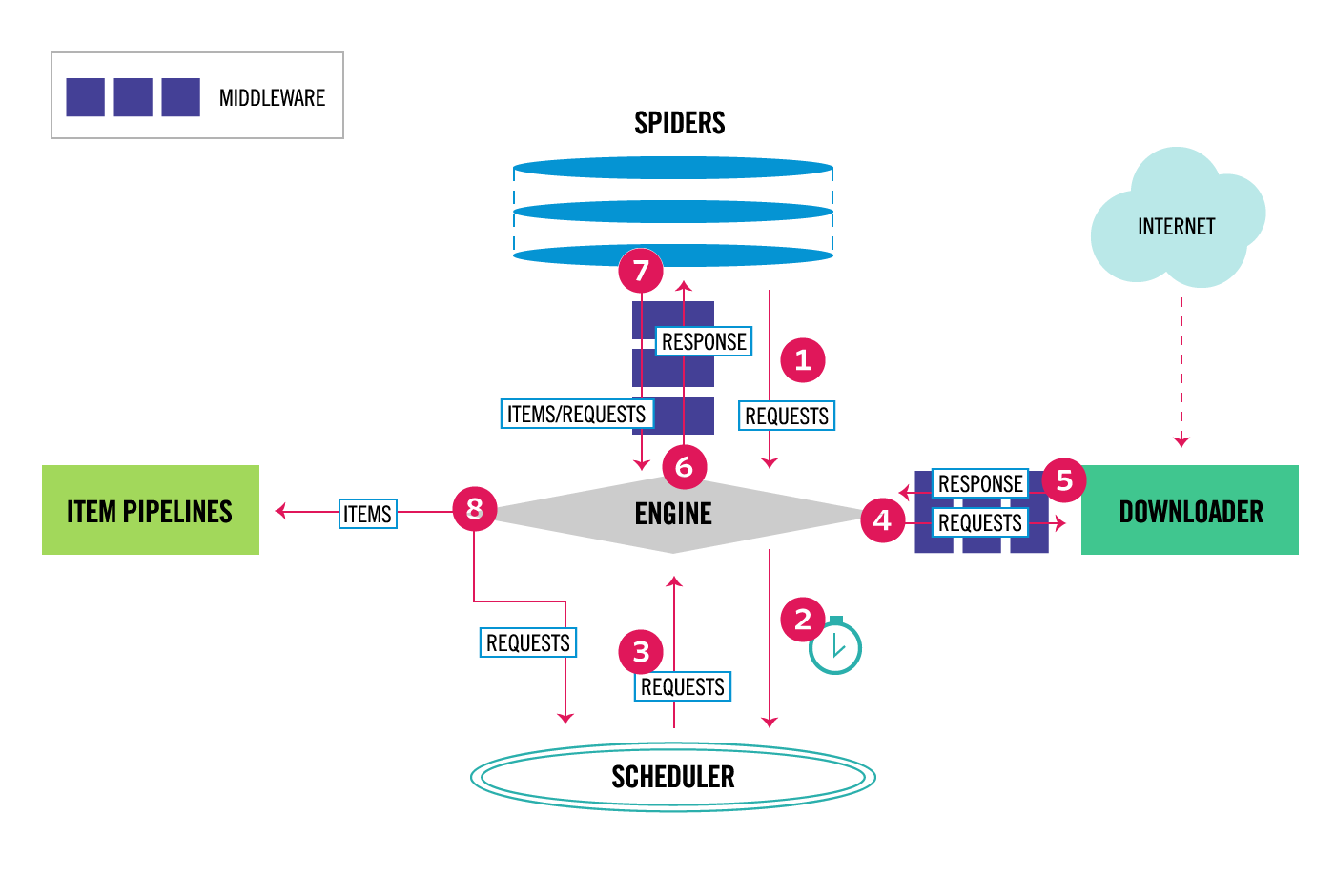
5.1 数据抓取

网络爬虫（web crawler），也叫网络蜘蛛（spider），是一种用来自动浏览万维网的网络机器人。其目的一般为编纂网络索引。

Scrapy一个开源和协作的框架，其最初是为了页面抓取 (更确切来说, 网络抓取 )所设计的，使用它可以以快速、简单、可扩展的方式从网站中提取所需的数据。但目前Scrapy的用途十分广泛，可用于如数据挖掘、监测和自动化测试等领域，也可以应用在获取API所返回的数据(例如 Amazon Associates Web Services ) 或者通用的网络爬虫。

Scrapy 是基于twisted框架开发而来，twisted是一个流行的事件驱动的python网络框架。因此Scrapy使用了一种非阻塞（又名异步）的代码来实现并发。

Scrapy框架结构如下



Scrapy框架各个组件的介绍:

1. 引擎(EGINE)

引擎负责控制系统所有组件之间的数据流，并在某些动作发生时触发事件.

2.调度器(SCHEDULER)

用来接受引擎发过来的请求, 压入队列中, 并在引擎再次请求的时候返回. 可以想像成一个URL的优先级队列, 由它来决定下一个要抓取的网址是什么, 同时去除重复的网址。

3. 下载器(DOWLOADER)

用于下载网页内容, 并将网页内容返回给EGINE，下载器是建立在twisted这个高效的异步模型上的

4.爬虫(SPIDERS)

SPIDERS是开发人员自定义的类，用来解析responses，并且提取items，或者发送新的请求

5. 项目管道(ITEM PIPLINES)

在items被提取后负责处理它们，主要包括清理、验证、持久化（比如存到数据库）等操作

6. 下载器中间件(Downloader Middlewares)

位于Scrapy引擎和下载器之间，主要用来处理从EGINE传到DOWLOADER的请求request，已经从DOWNLOADER传到EGINE的响应response，你可用该中间件做以下几件事。

7. 爬虫中间件(Spider Middlewares)

位于EGINE和SPIDERS之间，主要工作是处理SPIDERS的输入（即responses）和输出（即requests）。

5.2 数据存储

txt文本由于结构简单，文本文件被广泛用于记录信息。它能够避免其它文件格式遇到的一些问题。此外，当文本文件中的部分信息出现错误时，往往能够比较容易的从错误中恢复出来，并继续处理其余的内容。文本文件的一个缺点是，它的[熵](https://zh.wikipedia.org/wiki/%E7%86%B5_(%E4%BF%A1%E6%81%AF%E8%AE%BA)" \o "熵 (信息论))往往较低，也就是说，其实本可以用更小的存储空间记录这些信息。.txt是包含极少格式信息的文字文件的扩展名。.txt格式并没有明确的定义，它通常是指那些能够被系统[终端](https://zh.wikipedia.org/wiki/%E7%B5%82%E7%AB%AF" \o "终端)或者简单的[文本编辑器](https://zh.wikipedia.org/wiki/%E6%96%87%E6%9C%AC%E7%BC%96%E8%BE%91%E5%99%A8" \o "文本编辑器)接受的格式。任何能读取文字的程序都能读取带有.txt扩展名的文件，因此，通常认为这种文件是通用的、[跨平台](https://zh.wikipedia.org/wiki/%E8%B7%A8%E5%B9%B3%E5%8F%B0" \o "跨平台)的。

在英文文本文件中，ASCII字符集是最为常见的格式，而且在许多场合，它也是默认的格式。对于带重音符号的和其它的非ASCII字符，必须选择一种字符编码。在很多系统中，字符编码是由计算机的区域设置决定的。常见的字符编码包括支持许多欧洲语言的[ISO 8859-1](https://zh.wikipedia.org/wiki/ISO_8859-1" \o "ISO 8859-1)。

由于许多编码只能表达有限的字符，通常它们只能用于表达几种语言。[Unicode](https://zh.wikipedia.org/wiki/Unicode" \o "Unicode)制定了一种试图能够表达所有已知语言的标准，Unicode字符集非常大，它囊括了大多数已知的字符集。Unicode有多种字符编码，其中最常见的是[UTF-8](https://zh.wikipedia.org/wiki/UTF-8" \o "UTF-8)，这种编码能够向后兼容ASCII，相同内容的ASCII文本文件和UTF-8文本文件完全一致。

5.3 数据处理

数据（Data）是对事实、概念或指令的一种表达形式，可由人工或[自动化](https://baike.baidu.com/item/%E8%87%AA%E5%8A%A8%E5%8C%96" \t "_blank)装置进行处理。数据经过解释并赋予一定的意义之后，便成为信息。数据处理（data processing）是对数据的采集、[存储](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8/1582924" \t "_blank)、检索、加工、变换和传输。

数据处理的基本目的是从大量的、可能是杂乱无章的、难以理解的数据中抽取并推导出对于某些特定的人们来说是有价值、有意义的数据。

数据处理是系统工程和自动控制的基本环节。数据处理贯穿于社会生产和社会生活的各个领域。数据处理技术的发展及其应用的广度和深度，极大地影响了人类社会发展的进程。

1.停用词处理:

停用词是指在信息检索中，为节省存储空间和提高[搜索](https://baike.baidu.com/item/%E6%90%9C%E7%B4%A2/1806" \t "_blank)效率，在处理[自然语言](https://baike.baidu.com/item/%E8%87%AA%E7%84%B6%E8%AF%AD%E8%A8%80/4146019" \t "_blank)数据（或文本）之前或之后会自动过滤掉某些字或词，这些字或词即被称为Stop Words（停用词）。这些停用词都是人工输入、非自动化生成的，生成后的停用词会形成一个停用词表。但是，并没有一个明确的停用词表能够适用于所有的工具。甚至有一些工具是明确地避免使用停用词来支持短语搜索的。[Hans Peter Luhn](https://zh.wikipedia.org/w/index.php?title=Hans_Peter_Luhn&action=edit&redlink=1" \o "Hans Peter Luhn（页面不存在）), 作为信息检索的先驱者之一，他创造了这个短语并在他的研究中应用这个概念，为信息检索工作做出了贡献。

具体操作:

在GitHub上下载停用词: <https://github.com/fighting41love/funNLP>

下载后解压，把对应的文件导入到项目中，把该文件读取，作为一个列表

2.jieba分词:

Python中分分词工具很多，包括盘古分词、Yaha分词、Jieba分词、清华THULAC等。其中它们的基本用法都相差不大，但是Yaha分词不能处理如“黄琉璃瓦顶”或“圜丘坛”等词，所以使用了结巴分词。它主要有以下 3 种特性:

1. 支持 3 种分词模式：精确模式、全模式、搜索引擎模式

2. 支持繁体分词

3. 支持自定义词典

[jieba](https://github.com/fxsjy/jieba" \t "_blank) 是目前最好的 Python 中文分词组件，所有我们在本次实训中选择该库文件对数据进行处理。

5.4 数据可视化

数据可视化，是关于数据视觉表现形式的科学技术研究。其中，这种数据的视觉表现形式被定义为，一种以某种概要形式抽提出来的信息，包括相应信息单位的各种属性和变量。

它是一个处于不断演变之中的概念，其边界在不断地扩大。主要指的是技术上较为高级的技术方法，而这些技术方法允许利用图形、[图像处理](https://baike.baidu.com/item/%E5%9B%BE%E5%83%8F%E5%A4%84%E7%90%86/294902" \t "_blank)、[计算机视觉](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA%E8%A7%86%E8%A7%89/2803351" \t "_blank)以及用户界面，通过表达、建模以及对立体、表面、属性以及动画的显示，对数据加以可视化解释。与立体建模之类的特殊技术方法相比，数据可视化所涵盖的技术方法要广泛得多。

1.生成柱状图:

**matplotlib**是[Python](https://zh.wikipedia.org/wiki/Python" \o "Python)编程语言及其数值数学扩展包 [NumPy](https://zh.wikipedia.org/wiki/NumPy" \o "NumPy)的可视化操作界面。它利用通用的[图形用户界面工具包](https://zh.wikipedia.org/wiki/%E9%83%A8%E4%BB%B6%E5%B7%A5%E5%85%B7%E7%AE%B1" \o "部件工具箱)，如Tkinter, wxPython, [Qt](https://zh.wikipedia.org/wiki/Qt" \o "Qt)或[GTK+](https://zh.wikipedia.org/wiki/GTK%2B" \o "GTK+)，向应用程序嵌入式绘图提供了[应用程序接口](https://zh.wikipedia.org/wiki/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F%E6%8E%A5%E5%8F%A3)（API）。此外，matplotlib还有一个基于图像处理库（如开放图形库OpenGL）的pylab接口，其设计与[MATLAB](https://zh.wikipedia.org/wiki/MATLAB" \o "MATLAB)非常类似--尽管并不怎么好用。SciPy就是用matplotlib进行图形绘制。

Matplotlib是一个Python 2D绘图库，可以生成各种硬拷贝格式和跨平台交互式环境的出版物质量数据。Matplotlib可用于Python脚本，Python和IPython shell，Jupyter笔记本，Web应用程序服务器和四个图形用户界面工具包。

Matplotlib试图让简单易事的事情成为可能。你只需几行代码即可生成绘图，直方图，功率谱，条形图，误差图，散点图等。

对于简单的绘图，pyplot模块提供类似MATLAB的接口，特别是与IPython结合使用时。 对于高级用户，你可以通过面向对象的界面或通过MATLAB用户熟悉的一组函数完全控制线型，字体属性，轴属性等。

1. 生成词云

“词云”这个概念由美国西北大学新闻学副教授、新媒体专业主任里奇·戈登（Rich Gordon）于近日提出。戈登做过编辑、记者，曾担任迈阿密先驱报（Miami Herald）新媒体版的主任。他一直很关注网络内容发布的最新形式——即那些只有互联网可以采用而报纸、广播、电视等其它媒体都望尘莫及的传播方式。通常，这些最新的、最适合网络的传播方式，也是最好的传播方式。 因此，“词云”就是对网络文本中出现频率较高的“关键词”予以视觉上的突出，形成“关键词云层”或“关键词渲染”，从而过滤掉大量的文本信息，使浏览网页者只要一眼扫过文本就可以领略文本的主旨

词云图，也叫文字云，是对文本中出现频率较高的“关键词”予以视觉化的展现，词云图过滤掉大量的低频低质的文本信息，使得浏览者只要一眼扫过文本就可领略文本的主旨。

6．结论

对采集的一批样本进行关键字分析，我们可以选择TF-IDF，对所有的关键字进行词频统计，并绘制出关键字云图。

如果我们要分析一批样本中用户的分类，用户的行为，用户 的目的，我们可以选择TextRank抽取指定词性的关键字进行统计分析。

7．参考文献

1. 王仁武，python与数据科学.华东师范大学出版社
2. Jacqueline Kazil,Katharine Jarmul,Python数据处理 人民邮电出版社
3. 刘大成,Python数据可视化之matplotlib实践,电子工业出版社
4. 王斌会,多元统计分析及R语言建模 暨南大学出版社
5. [英)伯德](https://book.douban.com/search/%E4%BC%AF%E5%BE%B7)/[(英)克莱因](https://book.douban.com/search/%E5%85%8B%E8%8E%B1%E5%9B%A0)/[(美)洛普](https://book.douban.com/search/%E6%B4%9B%E6%99%AE) ，Python自然语言处理.东南大学出版社，2010