Python 入门

工作坊资料： https://github.com/OliverWuXY/AI-DH

吴小勇

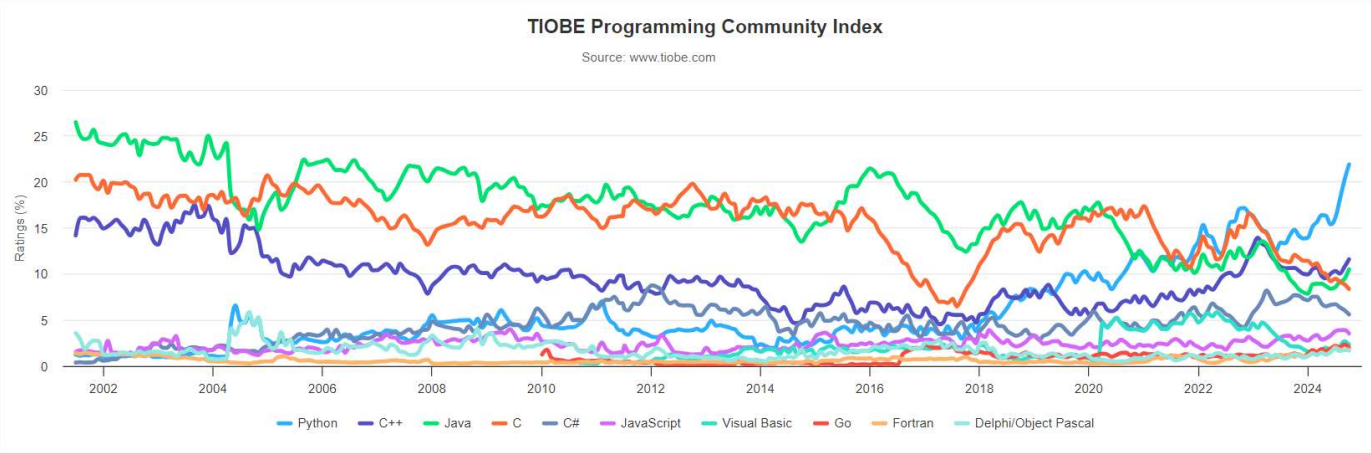
微信号：oliverwuxy

主要内容

- 为什么选择python
- 数字人文三个方面的应用
- Python基础

为什么选择python

Oct 2024	Oct 2023	Change	Programming Language	Ratings	Change
1	1		Python	21.90%	+7.08%
2	3	^	C++	11.60%	+0.93%
3	4	^	Java	10.51%	+1.59%
4	2	v	C	8.38%	-3.70%
5	5		C#	5.62%	-2.09%
6	6		JavaScript	3.54%	+0.64%
7	7		Visual Basic	2.35%	+0.22%
8	11	^	Go	2.02%	+0.65%
9	16	^^	Fortran	1.80%	+0.78%
10	13	^	Delphi/Object Pascal	1.68%	+0.38%
11	9	v	SQL	1.64%	-0.15%
12	14	^	MATLAB	1.48%	+0.22%
13	20	^^	Rust	1.45%	+0.53%
14	12	v	Scratch	1.41%	+0.05%
15	8	vv	PHP	1.21%	-0.69%
16	10	vv	Assembly language	1.13%	-0.51%
			R	1.09%	+0.12%
			Ruby	0.99%	+0.07%
			COBOL	0.99%	+0.23%
			Swift	0.98%	-0.09%



https://www.tiobe.com/tiobe-index/

为什么选择python

优点

- 简单，易学
- 高层语言
- 丰富的库
- 适合科学计算

缺点

- 单行语句和命令行输出问题
- 给初学者带来困惑：严格的缩进
- 运行速度慢

https://baike.baidu.com/item/Python/407313#2-2

Python之禅

英语 （自动检测） ⇄ **中文（简体）**

The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

Python 之禅》，蒂姆-彼得斯著

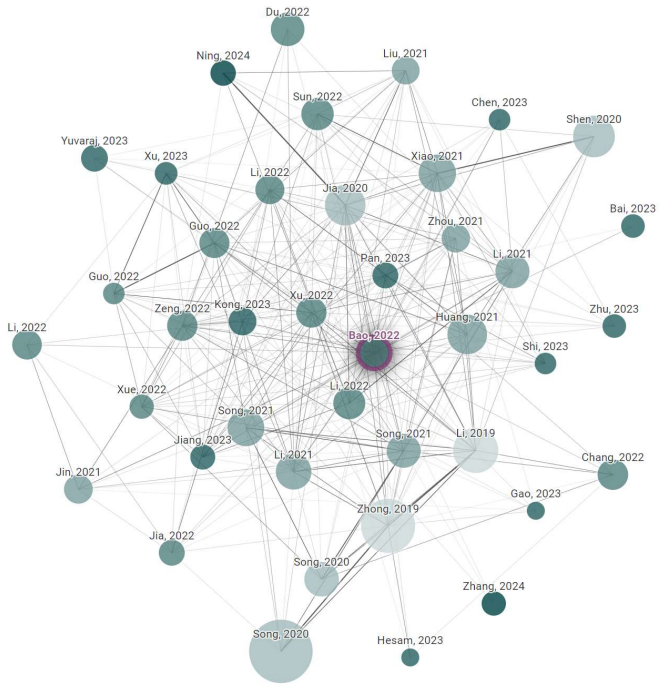
美丽胜过丑陋。
显式优于隐式。
简单胜于复杂。
复杂优于复杂。
扁平优于嵌套。
稀疏比密集好。
可读性很重要。
特殊情况不足以打破规则。
尽管实用性胜过纯粹性。
错误不应无声传递。
除非有明确规定。
面对模棱两可的情况，要拒绝猜测的诱惑。
应该有一种--最好只有一种--显而易见的方法。
尽管这种方法一开始可能并不明显，除非你是荷兰人。
现在做总比永远不做强。
尽管 “永远不做 “往往比 “现在不做 “要好。
如果实施起来很难解释，那就是个坏主意。
如果实现起来很容易解释，那可能是个好主意。
命名空间就是一个伟大的想法--让我们多做这样的事情吧！

快速学编程

- 技术在数字人文中扮演什么角色
- 找一本书从头到尾学习？
- 视频，step-by-step?
- 目标式
- 项目驱动
- 按需学习

网络分析

- 网络分析是一种多学科的方法,可以应用于社会网络、信息网络、生物网络等各个领域。
- 网络分析是一种从图论和复杂网络理论出发，利用数学和计算机科学的方法来研究和分析现实生活中的各种网络结构及其特征和规律的学科。
- 网络分析的研究对象可以是社交网络、互联网、生物网络等各种类型的网络。
- 网络分析的核心思想是通过对网络中的节点和边的分析，揭示出网络的拓扑结构、节点重要性和社区结构等信息，从而更好地理解和解释现实世界中的各种复杂系统。



网络分析- Game of Thrones

目的：Character Interaction Networks for the HBO Series "Game of Thrones"

There are five interaction types. Character A and Character B are connected whenever:

- Character A speaks directly after Character B
- Character A speaks about Character B
- Character C speaks about Character A and Character B
- Character A and Character B are mentioned in the same stage direction
- Character A and Character B appear in a scene together

Game of Thrones数据来源
https://github.com/mathbeveridge/gameofthrones

了解数据-数据格式

- got-s1-edges.csv
- got-s1-nodes.csv
- got-s2-edges.csv
- got-s2-nodes.csv
- got-s3-edges.csv
- got-s3-nodes.csv
- got-s4-edges.csv
- got-s4-nodes.csv
- got-s5-edges.csv
- got-s5-nodes.csv
- got-s6-edges.csv
- got-s6-nodes.csv
- got-s7-edges.csv
- got-s7-nodes.csv
- got-s8-edges.csv
- got-s8-nodes.csv

	A	B	C	D
1	Source	Target	Weight	Season
2	NED	ROBERT	192	1
3	DAENERYS	JORAH	154	1
4	JON	SAM	121	1
5	LITTLEFINGER	NED	107	1
6	NED	VARYS	96	1
7	DAENERYS	DROGO	91	1
8	ARYA	NED	90	1
9	CATELYN	ROBB	90	1
10	BRONN	TYRION	86	1

	A	B
1	Id	Label
2	ADDAM_M	Addam
3	AEGON	Aegon
4	AERYS	Aerys
5	ALLISER_TH	Allister
6	ARYA	Arya
7	ASSASSIN	Assassin
8	BAELOR	Baelor
9	BALON	Balon
10	BARRISTAN	Barristan

了解数据-加载数据、数据处理

```
import pandas as pd
# load data
ProjDir = "E:/PythonProj/AI_DH/network-analysis/"
df = pd.read_csv(ProjDir + "data/gameofthrones-master/got-s1-edges.csv")
orginalDf = df

print(df.head)
print(df.columns)
# 数据表头
# Source,Target,Weight,Season
# pick only important weights (hard threshold)
df = df.loc[df['Weight']>20, :]
# print(df)
```

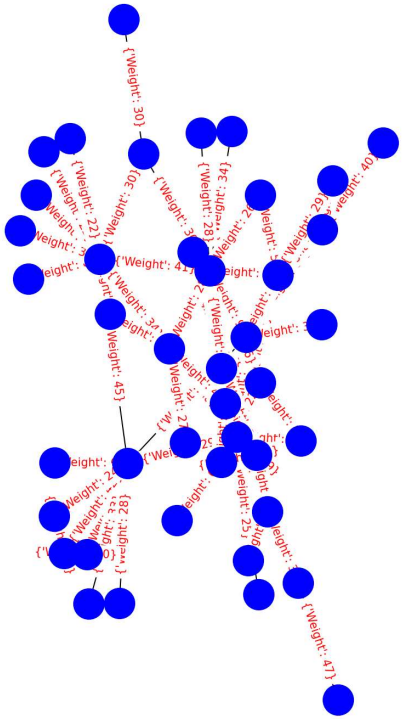
数据建模

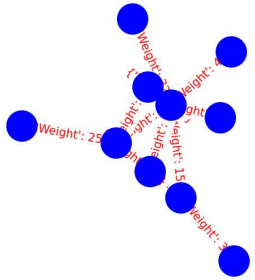
```
import networkx as nx
# load pandas df as networkx graph
G = nx.from_pandas_edgelist(df,
                            source='Source',
                            target='Target',
                            edge_attr='Weight')
print("Number of unique characters:", len(G.nodes))
print("Number of connections:", len(G.edges))
```

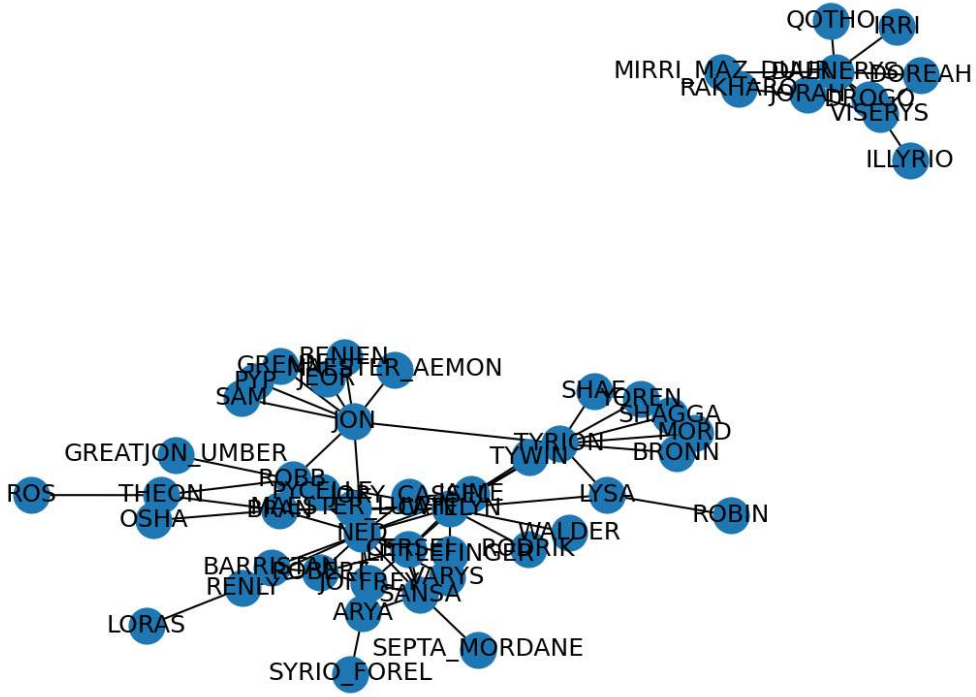
分析

```
# 度中心性
# 度中心性是衡量网络中节点重要性的指标。它只是连接到节点的边的数量，由节点的最大可能度标准化。
degree_centrality = nx.degree_centrality(G)
print(degree_centrality)

# 最短路径
# 寻找两个节点之间的最短路径是图论中常见的问题。NetworkX 提供了几个用于计算最短路径的函数，例如 shortest_path() 和 shortest_path_length()：
path = nx.shortest_path(G, source="LYSA", target="JEOR")
length = nx.shortest_path_length(G, source="LYSA", target="LYSA")
print(f"Shortest path: {path}, Length: {length}")

# 聚类系数
# 聚类系数是衡量图中节点形成簇或紧密结合的组的趋势的指标。它是连接到节点的三角形数量与可能连接到该节点的三角形数量的比率。可以使用
# clustering()
# 函数计算图中所有节点的聚类系数：

clustering_coefficient = nx.clustering(G)
print(clustering_coefficient)

# 社区检测
# 社区检测是在图中查找节点组的过程，这些节点组之间的连接比与网络其他部分的连接更紧密。NetworkX 提供了几种社区检测算法，例如 Louvain 方法和 Girvan-Newman 方法
from networkx.algorithms import community
communities = list(community.greedy_modularity_communities(G))
print("There are {} communities.".format(len(communities) ))
print(communities)
```

可视化分析-NetworkX



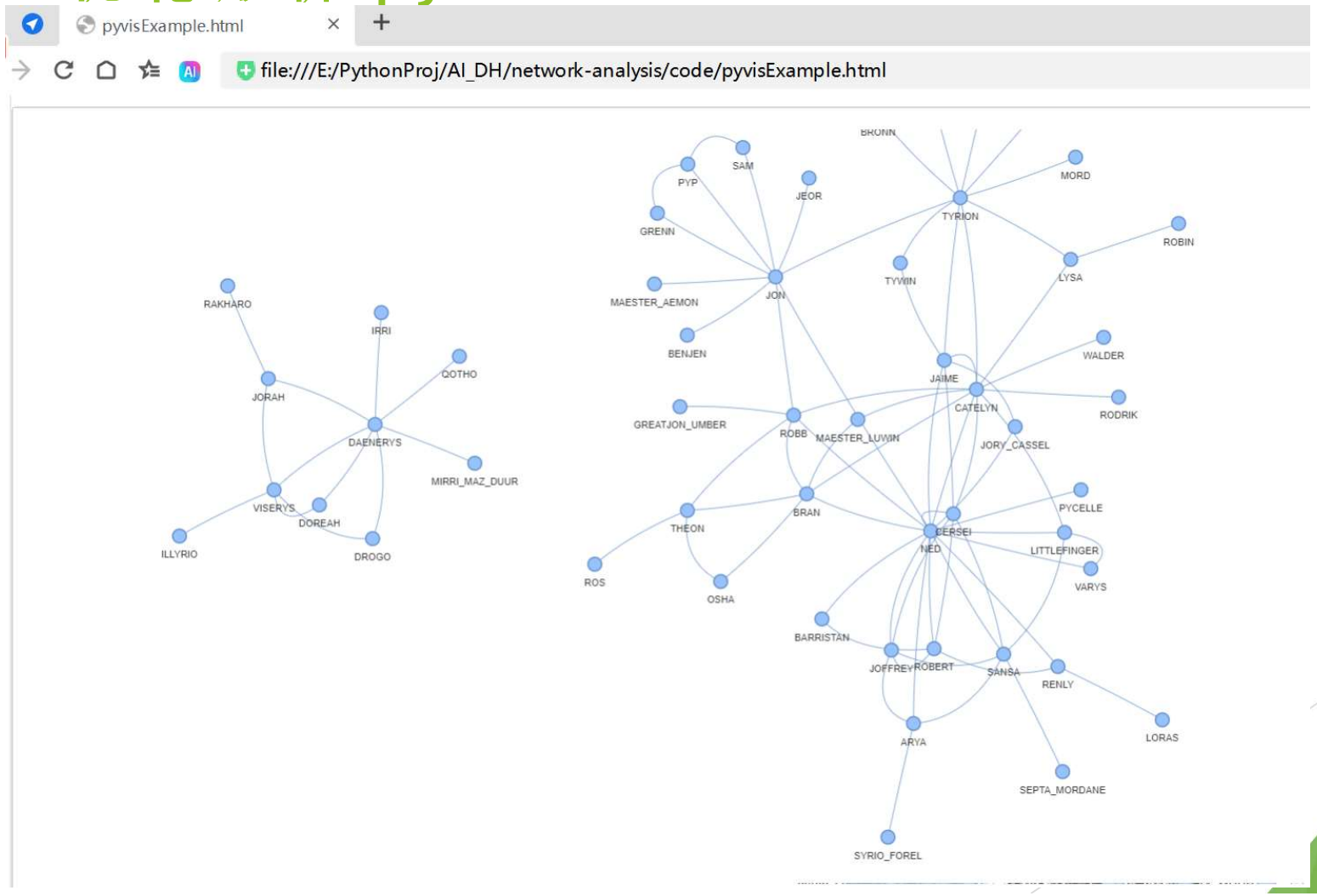




NetworkX 可视化优缺点

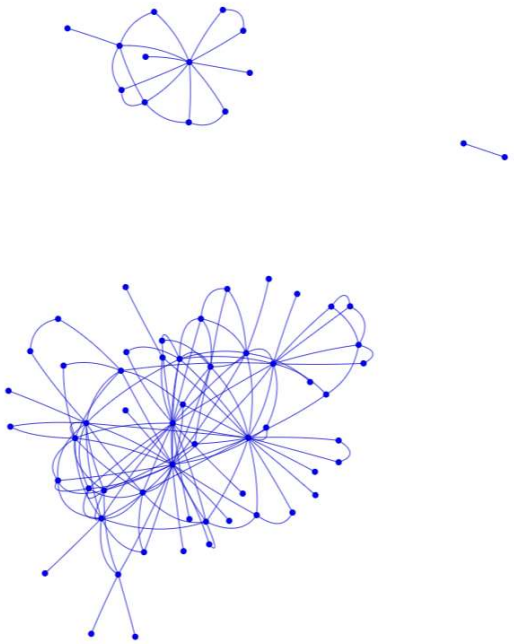
- 优点：
  - 内置绘图模块
  - 大量自定义的绘图选项
  - 网络指标算法比较丰富
- 缺点：
  - 固定网络，不能交互
  - 处理的网络规模有限

可视化分析-pyvis

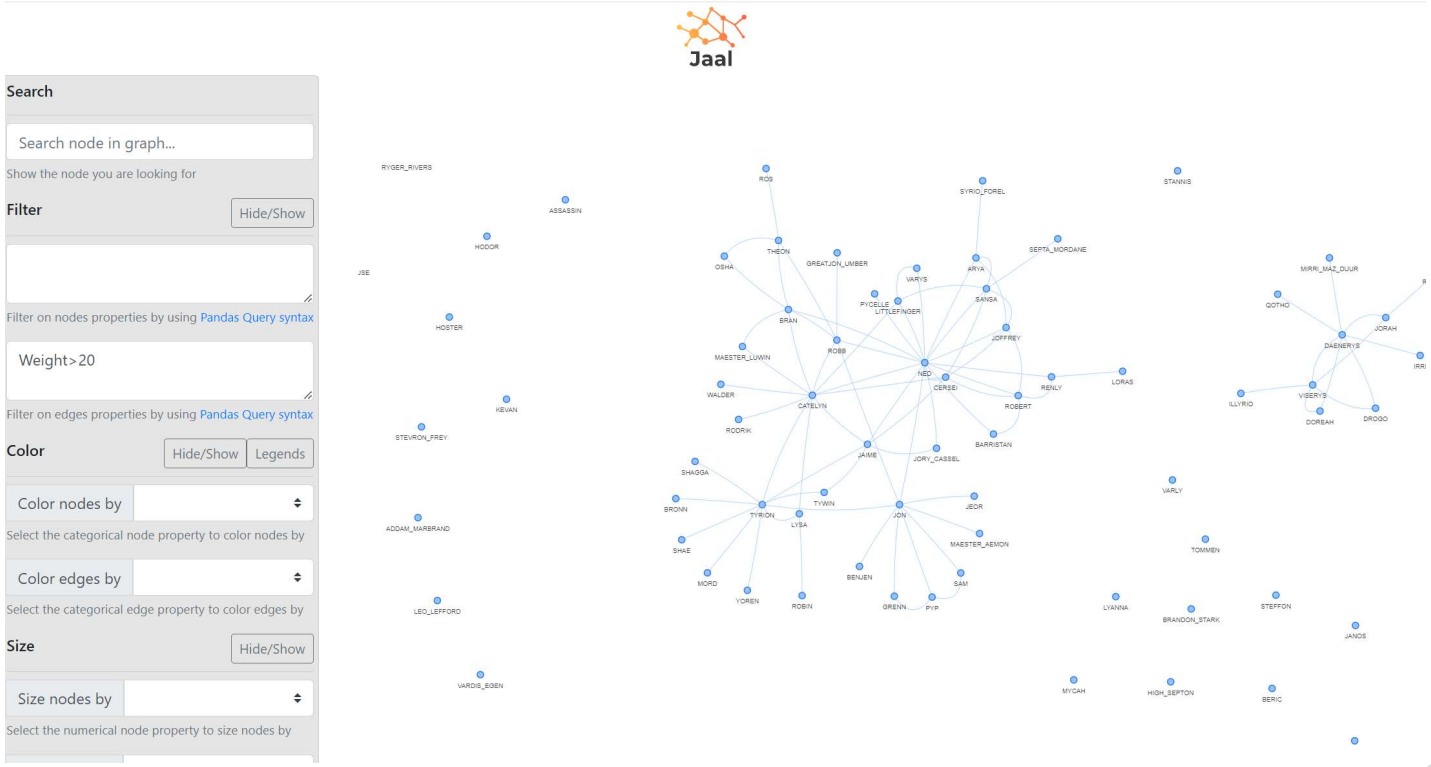

pyvisExample.html
file:///E:/PythonProj/AI_DH/network-analysis/code/pyvisExample.html
RAKHARO
IRRI
QOTHO
JORAH
DAENERYS
MIRRI_MAZ_DUUR
VISERYS
DOREAH
ILLYRIO
DROGO
BRONN
SAM
PYP
JEOR
MORD
TYRION
GRENN
ROBIN
TYWIN
LYSA
MAESTER_AEMON
JON
BENJEN
WALDER
JAIME
CATELYN
RODRIK
GREATJON_UMBER
ROBB
MAESTER_LUWIN
JORY_CASSEL
PYCELLE
THEON
BRAN
CERSEI
NED
LITTLEFINGER
ROS
OSHA
VARYS
BARRISTAN
JOFFREY
ROBERT
SANSA
RENLY
LORAS
ARYA
SEPTA_MORDANE
SYRIO_FOREL

```html
<html>
    <head>
        <meta charset="utf-8">

            <script src="lib/bindings/utils.js"></script>
            <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/vis-network/9.1.2/dist/dist/vis-network.m
            <script src="https://cdnjs.cloudflare.com/ajax/libs/vis-network/9.1.2/dist/vis-network.min.js" integrity="sha


<center>
<h1></h1>
</center>

<!-- <link rel="stylesheet" href="../node_modules/vis/dist/vis.min.css" type="text/css" />
<script type="text/javascript" src="../node_modules/vis/dist/vis.js"> </script>-->
        <link
          href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css"
          rel="stylesheet"
          integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
          crossorigin="anonymous"
        />
        <script
          src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js"
          integrity="sha384-JEW9xMcG8R+pH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf"
          crossorigin="anonymous"
        ></script>


        <center>
          <h1></h1>
        </center>
        <style type="text/cssl">

            #mynetwork {
                width: 1000;
                height: 600;
                background-color: #ffffff;
                border: 1px solid lightgray;
                position: relative;
                float: left;
            }





        </style>
    </head>


    <body>
        <div class="card" style="width: 100%">


            <div id="mynetwork" class="card-body"></div>
        </div>
```

可视化分析-Visdcc in Dash



可视化分析-Jaal

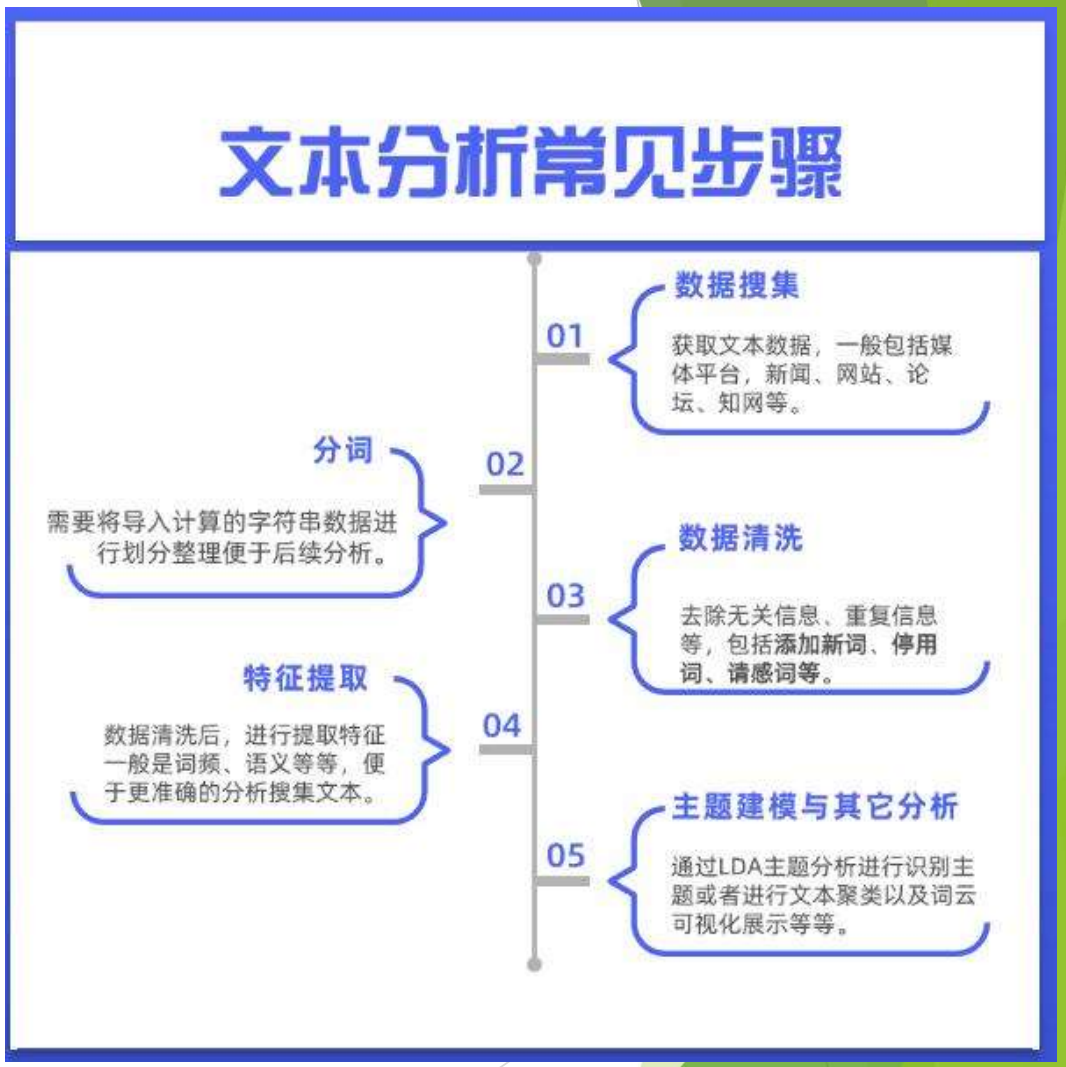

Jaal
Search
Search node in graph...
Show the node you are looking for
Filter
Hide/Show
Filter on nodes properties by using Pandas Query syntax
Weight>20
Filter on edges properties by using Pandas Query syntax
Color
Hide/Show
Legends
Color nodes by
Select the categorical node property to color nodes by
Color edges by
Select the categorical edge property to color edges by
Size
Hide/Show
Size nodes by
Select the numerical node property to size nodes by
RYGER_RIVERS
ASSASSIN
HODOR
HOSTER
KEVAN
STEVRON_FREY
ADDAM_MARBRAND
LEO_LEFFORD
VARDIS_EGEN
ROS
SYRIO_FOREL
STANNIS
THEON
OSHA
GREATJON_UMBER
VARYS
ARYA
SEPTA_MORDANE
PYCELLE
LITTLEFINGER
SANSA
BRAN
JOFFREY
ROBB
MAESTER_LUWIN
NED
CERSEI
RENLY
LORAS
WALDER
CATELYN
ROBERT
RODRIK
BARRISTAN
JAIME
JORY_CASSEL
SHAGGA
BRONN
TYRION
TYWIN
LYSA
JON
JEOR
SHAE
MAESTER_AEMON
MORD
BENJEN
SAM
YOREN
ROBIN
GRENN
PYP
MIRRI_MAZ_DUUR
QOTHO
JORAH
DAENERYS
IRRI
ILLYRIO
VISERYS
DOREAH
DROGO
VARLY
TOMMEN
LYANNA
BRANDON_STARK
STEFFON
JANOS
MYCAH
HIGH_SEPTON
BERIC

谈论

- 根据上一节内容，总结一下做一个人文分析的项目的基本过程。
- 基于以上过程，你觉得还需要考虑什么问题？
- 什么项目是一个好的数字人文项目？

文本分析

- 文本分析作为自然语言处理应用领域之一，在日常的工作中使用广泛，随着近几年短视频等新媒介的爆火，基于文本的用户意图识别、情感分析对于企业中产品或者服务的流程优化、用户需求分析、潜在用挖掘户等，都起到举足轻重的作用。
- 对于学术呢？



https://baijiahao.baidu.com/s?id=1792137971267077650&wfr=spider&for=pc

https://zhuanlan.zhihu.com/p/690597225

文本分析

- 用户字典
- 分词
- 地理解析(Geoparsing)

文本分析—用户字典

- 生成用户字典
- 使用用户字典

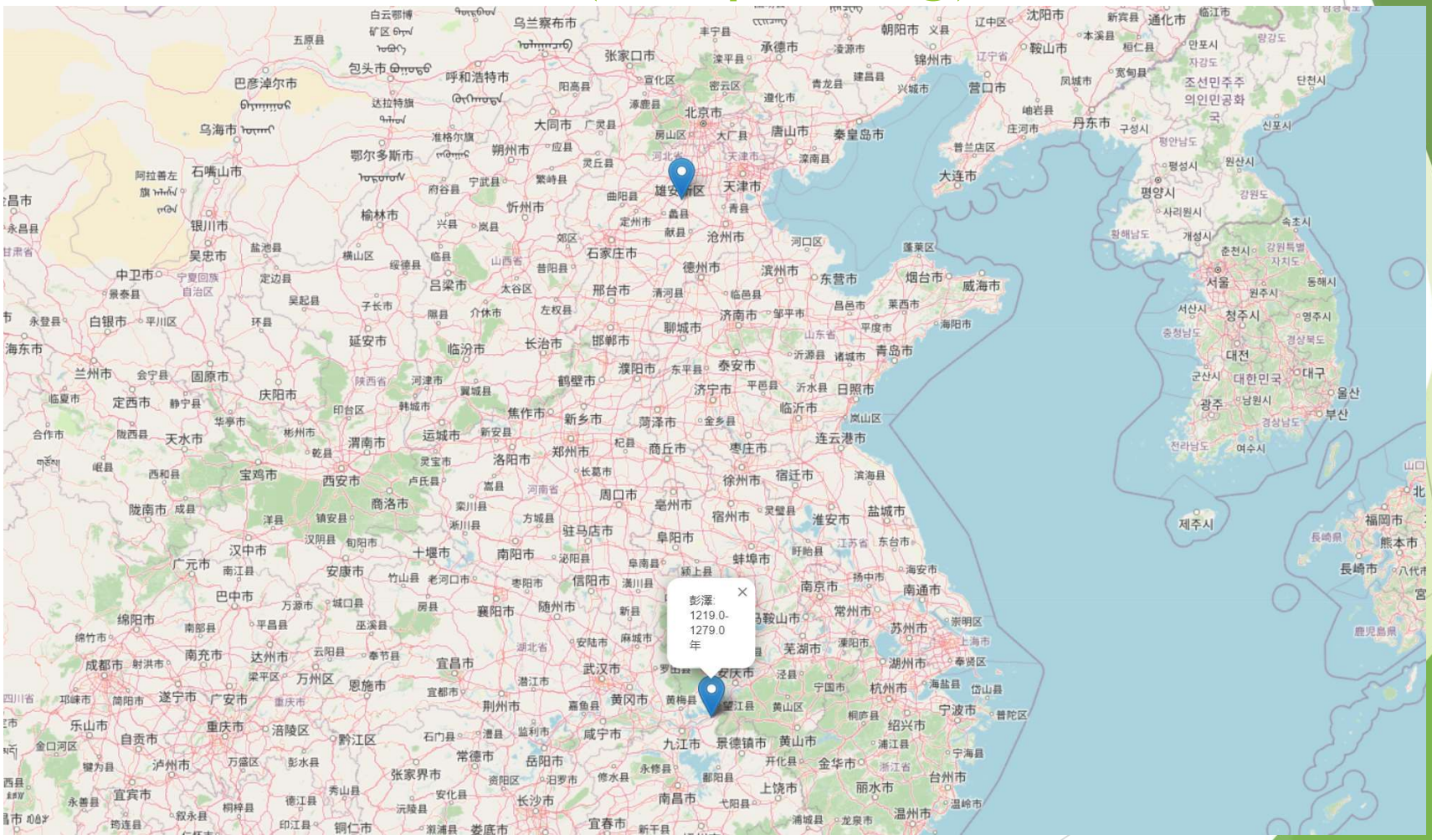
```python
'''
用户字典，地名
'''
placenames = []
with open('../data/placenames.txt', 'r', encoding='utf-8') as f:
    txt = f.read()

placenames = txt.split('\n')
print('用户地名库收录了{}个地名'.format(len(placenames)))

1 usage  OliverWuXY
def is_placename(placename):
    '''
    判断是否是地名
    :param placename:
    :return:
    '''
    if placename in placenames:
        return True
    else:
        return False
```

```python
'''
提取地方名
'''
import jieba
jieba.load_userdict("../data/placenames.txt") # 用户字典
# 解决jieba分词 load_userdict 加载自定义词库太慢的问题 https://blog.csdn.net/qq_29202513/article/details/85236995
text = "清趙世震修，汪澤延纂。世震遼東人，康熙間漢陰知縣。澤延漢陰縣人，貢生。考漢陰縣志創於成化十六年知縣張大綸，萬歷四十七年知縣張啟蒙，
words = [item for item in jieba.cut(text)]
places = [p for p in words if is_placename(p)]
print(places)
setPlaces = set(places)
print(setPlaces)
```

文本分析-分词

```python
import jieba
# jieba中的paddle模式
# https://blog.csdn.net/nkufang/article/details/129788741
import paddle

paddle.enable_static()
jieba.enable_paddle()# 启动paddle模式。 0.40版之后开始支持，早期版本不支持
strs=["我来到北京师范大学","刀郎的门票卖完了","北京师范大学珠海校区", "清滕天绶纂修天绶辽阳人荫生由广东潮州府同知升汉中府知府"]
for str in strs:
    seg_list = jieba.cut(str,use_paddle=True) # 使用paddle模式
    print("Paddle Mode: " + '/'.join(list(seg_list)))

seg_list = jieba.cut("我来到北京师范大学", cut_all=True)
print("Full Mode: " + "/ ".join(seg_list))  # 全模式

seg_list = jieba.cut("我来到北京师范大学", cut_all=False)
print("Default Mode: " + "/ ".join(seg_list))  # 精确模式

seg_list = jieba.cut("我来到北京师范大学")  # 默认是精确模式
print(", ".join(seg_list))

seg_list = jieba.cut_for_search("小明硕士毕业于北京师范大学文理学院，后在剑桥大学深造")  # 搜索引擎模式
print(", ".join(seg_list))
```

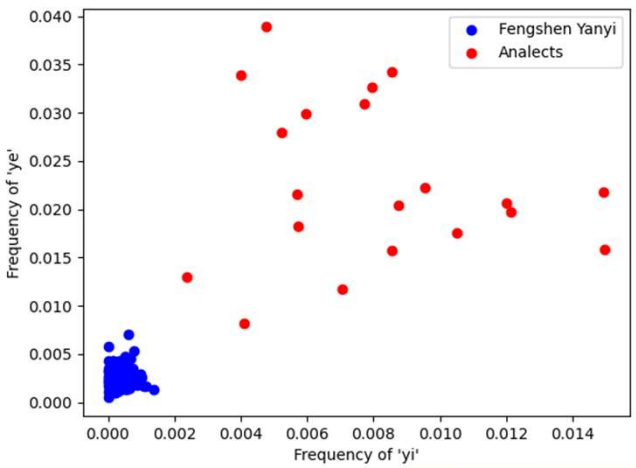
文本分析-地理解析(Geoparsing)

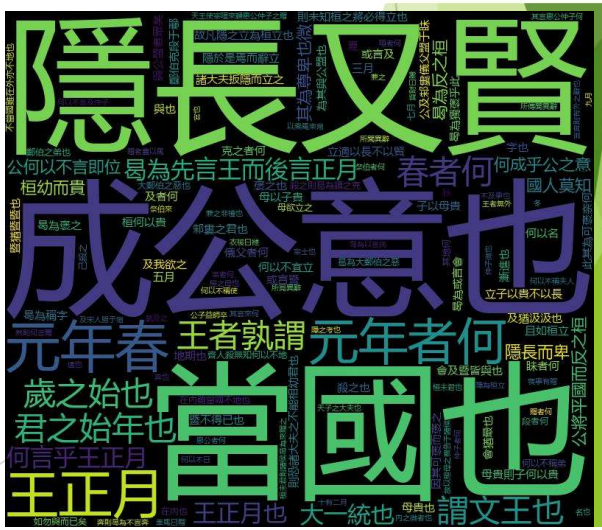


中国哲学书电子化计划Chinese Text Project

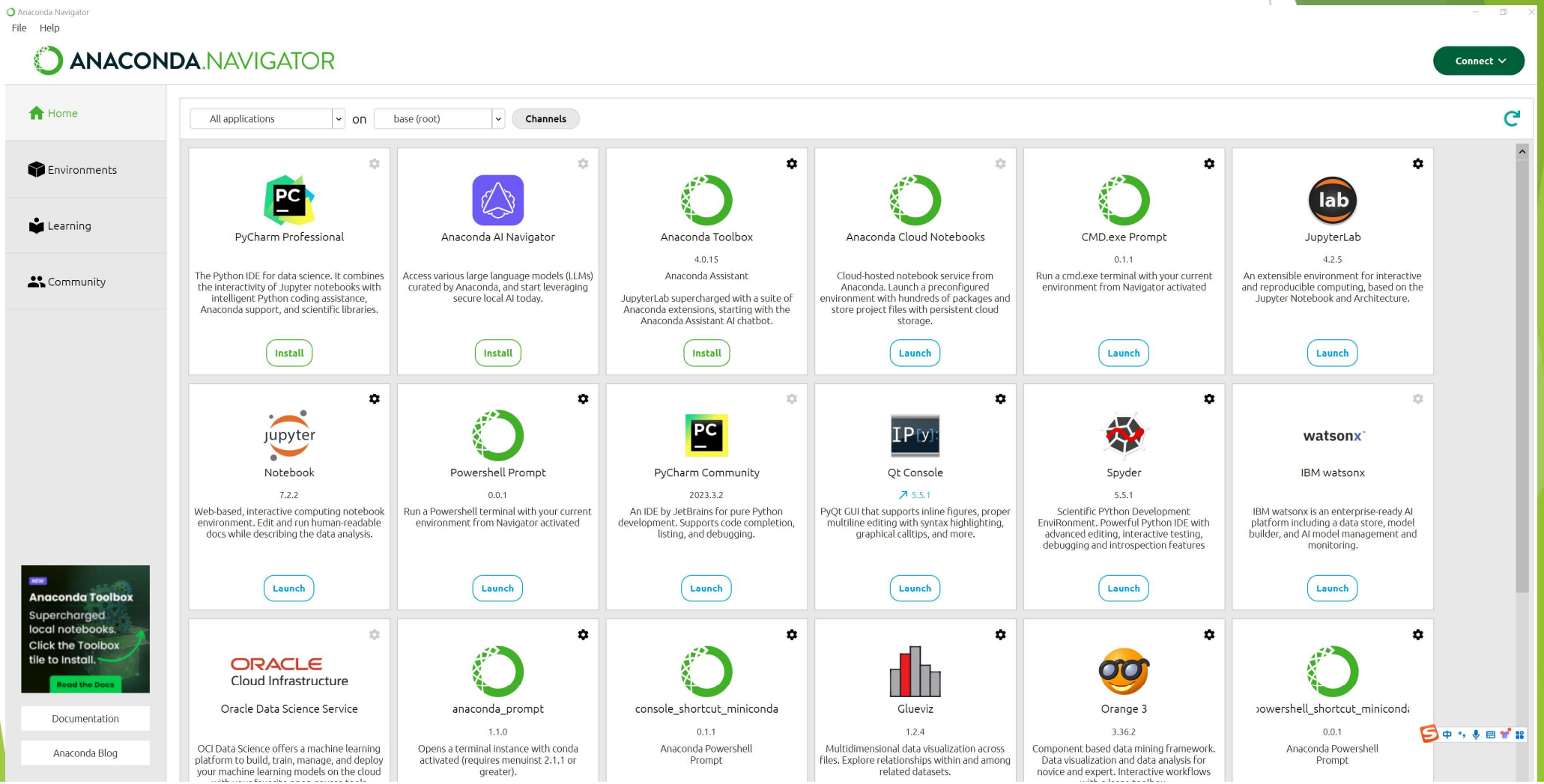
- 用Python了解Chinese Text Project
- 样例

1. 找出最长的段
2. 词频分析
3. 词云





Anaconda 环境


Anaconda Navigator
File Help
ANACONDA.NAVIGATOR
Connect
Home
Environments
Learning
Community
All applications
on
base (root)
Channels
PyCharm Professional
The Python IDE for data science. It combines the interactivity of Jupyter notebooks with intelligent Python coding assistance, Anaconda support, and scientific libraries.
Install
Anaconda AI Navigator
Access various large language models (LLMs) curated by Anaconda, and start leveraging secure local AI today.
Install
Anaconda Toolbox
4.0.15
Anaconda Assistant
JupyterLab supercharged with a suite of Anaconda extensions, starting with the Anaconda Assistant AI chatbot.
Install
Anaconda Cloud Notebooks
Cloud-hosted notebook service from Anaconda. Launch a preconfigured environment with hundreds of packages and store project files with persistent cloud storage.
Launch
CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated
Launch
JupyterLab
4.2.5
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch
Notebook
7.2.2
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch
Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated
Launch
PyCharm Community
2023.3.2
An IDE by JetBrains for pure Python development. Supports code completion, listing, and debugging.
Launch
Qt Console
5.5.1
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch
Spyder
5.5.1
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch
IBM watsonx
IBM watsonx is an enterprise-ready AI platform including a data store, model builder, and AI model management and monitoring.
Launch
Oracle Data Science Service
OCI Data Science offers a machine learning platform to build, train, manage, and deploy your machine learning models on the cloud
anaconda_prompt
1.1.0
Opens a terminal instance with conda activated (requires menuinst 2.1.1 or greater).
console_shortcut_miniconda
0.1.1
Anaconda Powershell Prompt
Glueviz
1.2.4
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Orange 3
3.36.2
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows
0.0.1
Anaconda Powershell Prompt
NEW
Anaconda Toolbox
Supercharged local notebooks. Click the Toolbox tile to install.
Read the Docs
Documentation
Anaconda Blog

Python基础

- 基本输入与输出
- 变量与常量
- 程序基本结构（顺序，分支，循环）
- 数据类型（基本数据类型，组合数据类型）
- 基本运算
- 函数
- 面向对象编程（相对难，开始可以不考虑）