求职查询系统

隐藏我们的个人信息：由：姚，陆，陈，曹

个人贡献：

姚：整个查询系统的数据处理、算法优化和UI开发

Lu：制作演示文稿幻灯片，安排我们的项目步骤

陈：爬行数据，逆数据类型和描述数据集

曹：登录系统的UI，查找参考，介绍及相关工作

摘要

对于毕业生来说，找工作往往是一件麻烦但必须做的事情。 这导致需要筛选工作招聘。 在这方面，信息检索可以发挥很大的作用。 我们的项目功能类似于搜索引擎，也有学习用户选择的功能，帮助筛选出对用户有用的信息。 我们将删除停止词与nltk的英语和哈尔滨工业大学停止词表的中文。 由于余弦相似度对排序的时间效率很高，因此将采用流排序算法对余弦相似度的结果进行排序。 此外，我们希望在现有的程序中，可以提高整体的运行速度，并将与大数据和机器学习相关的内容应用到项目中，从而达到使程序整体更加智能化的目的。

1. 导言
2. 我们问题的定义

我们的项目叫猎头。 显然，用户可以使用此应用程序来搜索作业。 换句话说，这个应用程序是为用户找到合适的工作而设计的。 我们将充分利用我们在信息检索中所学到的知识，结合我们已经知道的Python用法，构建一个搜索引擎。 作为一个搜索引擎，它的功能类似于百度和谷歌等大型搜索引擎。 并且我们的项目更专业的满足求职者的需求。

1. 我们问题的例子

举个例子，如果一个主修CS的人想成为JAVA工程师。 他需要做的就是输入关键词“JavaEngineer”。 我们的程序将在数据库中查询和显示包含这两个关键字的数据，这样他就可以方便、快速地获得他想要的工作信息。 除了直接搜索工作。 也许有些人想找一份离家近的工作，或者有些人想在另一个城市工作。 候选人也可以进入他们想在我们的搜索引擎工作的区域，我们将在这个城市提供一些替代的工作信息。 我们的项目还增加了一个学习功能，它旨在允许用户对被查询的信息作出判断。 如果查询到的信息与用户想要的内容无关，用户可以点击无关按钮。 并在进一步的查询工作中，排除无关信息。 当然，在我们的应用程序中也设计了相关的按钮。 当用户找到合适的工作信息时，可以点击相关按钮。 此消息的内容由系统记录和学习，在下一次查询中，系统对已记录的信息施加权重。 举个例子，有人在南京查询工作信息，他点击相关的按钮，招聘信息需要一个Java工程师。 当他第二次查询时，会出现更多关于Java工程师的信息。

1. 动机

在传统的求职过程中，应聘者与招聘人员之间的沟通渠道太少。 很多时候，候选人往往花很多时间没有找到合适的工作，许多公司也没有为此雇用合适的人。 因此，我们的项目的价值是节省候选人寻找正确工作机会的时间。

1. 申请

目前，一些成熟的网站或应用程序也提供了类似的功能，我们在项目的设计中借用了它们的一些设计

利品平台为中高端人才提供岗位信息。 许多猎头为你提供在线求职服务。 涵盖多个行业知名企业，提供简历指导，面试指导等求职服务。

1. 相关工程

首先，我们查了一些关于求职的文件。 在“求职者图书馆指南”[1]的文章中，作者描述了在大多数公共图书馆中找到的职业资源。 包括有关Reader索引的信息和可用的特定引用。 在“求职者的资料手册”[2]，作者给出了员工找到工作所需要的意见。 这两篇文章使我们从根本上认识到什么是就业导向和其他求职资源，求职过程和候选人在寻找工作[3][4]时的考虑。

然后，我们查找了一些关于信息检索和搜索引擎的信息。 通过查阅这些资料，我们对相关的开发过程有了更深的了解，学会了一些处理数据的新方法。 例如，在文章“作业搜索引擎和使用方法”[5]中，作者提供了一种积累、处理和分类在线作业列表的方法，以供用户搜索。 作业列表是自动的，更有效的分类，允许用户更快和更容易地搜索作业列表。 在“对一个非常大的网络搜索引擎查询日志的分析”[6]中，他们对Alta Vista搜索引擎查询日志进行了分析，该查询日志由大约10亿条搜索请求条目组成，为期6周。 他们的相关分析表明，最高度相关的项目是短语的成分。 这一结果表明，搜索引擎将搜索词视为短语的一部分可能是有用的，即使用户没有明确地将它们指定为短语。 这些文章对我们有很大的帮助。

1. 数据集描述
2. 数据收集的方法

我们使用一种名为webspider的技术在特定的网站上爬行数据。 我们将这项技术分为五个步骤，并用Python实现。 一般这五个步骤如下部分所示。

首先，我们需要找到我们需要的网站的给定URL。 第二，我们访问网站，检查HTML源代码。 第三，我们使用工具网站将HTML代码解析为JSON类型，方便我们找到我们想要获得的数据。 第四，我们编写Python代码来模拟一个真实的人类向网站服务器发送请求，以获取数据。 最后，我们保存我们爬进csv文件的数据，并将其转换成utf-8类型。

我们将用一些图像详细地讨论这些步骤。 首先，我们从招聘网站“拉古王”获得数据集”。 这个网站允许我们在不登录的情况下获取数据。 方便我们获得Chrome网站的URL。 然后，我们访问网站的主页，然后找到Ajax位置。 为了方便地检查数据类型，我们将其解析为JSON类型。 我们检查网站的源代码，找到我们所需位置Ajax的响应，如图1所示，然后将响应代码放入JSON分析工具中，以获得数据类型，如下图所示的图像。

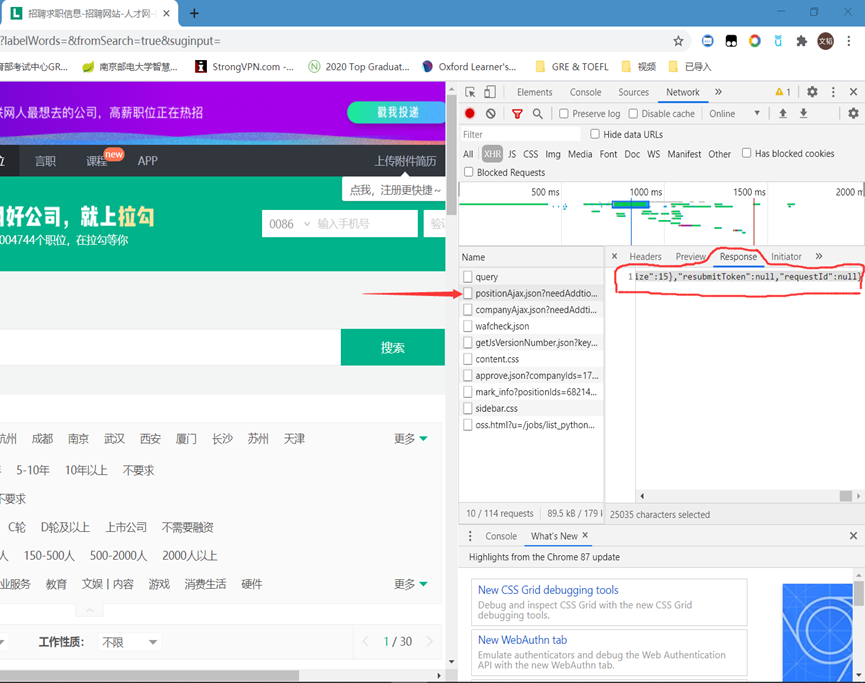


图1。 位置Ajax的响应



图2。 JSON解析后的数据类型结果

随后，我们创建了一个请求类会话来模拟用户向服务器发送请求，因为lagouwang使用异步加载来加载数据。 我们需要复制位置Ajax响应的请求URL。 最后，我们可以得到保存在csv文件中的数据。

1. 数据收集过程中遇到的问题

当我们开始爬行数据时，我们应该注意每个爬行的时间间隔。 我们在开始时将时间间隔设置为一个非常小的值，这样我们的请求就会被网站服务器拒绝。 因此，我们设置了一秒钟来爬取数据，以避免经常访问网站服务器而被拒绝。

1. 数据注释

因为我们将数据类型设置为utf-8，所以csv文件中充满了杂乱的代码。 为了解决这个问题，我们使用EXCEL工具将utf-8数据更改为可读内容。

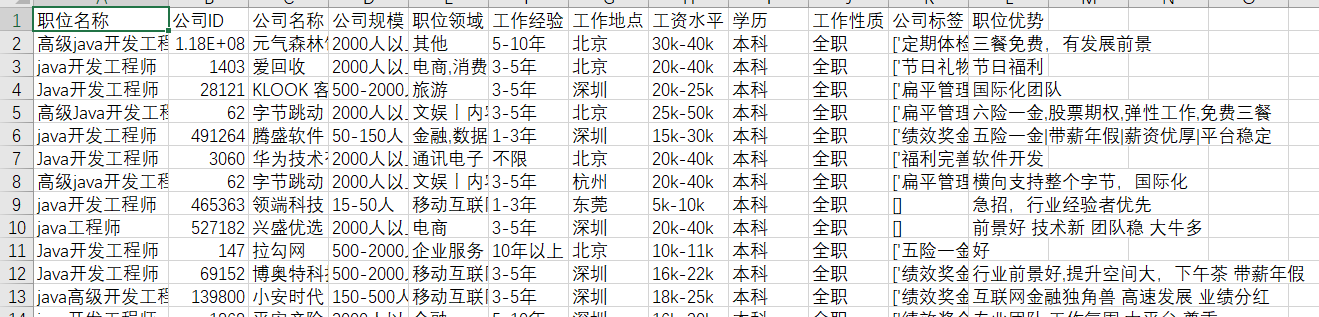
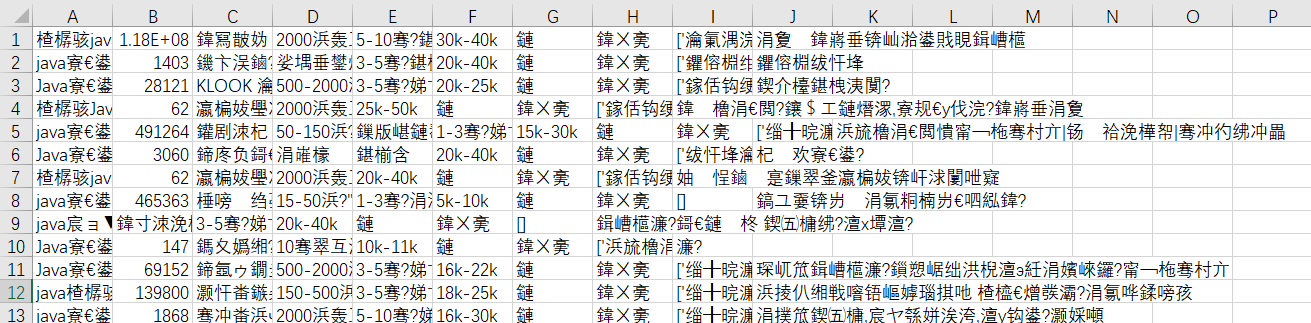


图3。 原始数据和转换后的数据

1. 方法说明

我们遵循基本的管道，并实现了排名搜索引擎与一些经典的算法，我们已经研究过。

首先，我们将每个记录中的元素组合成字符串，并将这些字符串作为文档。 然后我们标记它们。 中文文本需要通过分词来获得单个单词，所以我们导入jieba库来对我们从抓取数据的结果中得到的中文数据进行标记。 杰巴库是一个优秀的中文分词第三方库。 解巴库分词原理：用一个汉语词库来确定汉字之间的相关概率，汉字之间概率较高的单词形成短语形成分词结果。

杰巴库提供了三种分割模式：1)精确模式，返回没有冗余的列表式分词结果。 2)Full模式，返回具有冗余的列表类型分词结果。 3)搜索引擎模式返回一个列表类型的分词结果与冗余，细分较长的单词。 在这里我们选择精确的模式。 此外，我们还使用nltk来标记英语数据，如果有的话。 然后用nltk删除英文的stopword，哈尔滨工业大学中文的stopword列表。 为了将具有相同含义的单词结合起来，对于英语，我们在nltk中使用porterstemmer；对于汉语，HITIR-LabTongyiciCilin(Extended)可以在未来使用，但在这里我们不做汉语同义词合并。 然后，我们将每个文档中单词的频率计数成一个频率矩阵。 接下来，我们用归一化计算tfidf矩阵。 tfidf矩阵是tf矩阵和idf矩阵的乘法结果。 计算tf矩阵为，其中f为频率矩阵。 以国防军是关于整个语料库的，计算为，其中N是所有文档的数量，分母是包含这个术语的文档数量。

我们为数据库存储tfidf矩阵，并读取我们的搜索引擎每次启动，所以除了第一次，我们不必计算它。 计算需要时间。 以色列国防军也储存起来进行后续计算。

用户标记为相关或无关的查询和记录首先与记录相同，计算tfidf，但对整个语料库使用存储的IDF。 然后用标准rocchio公式：，生成一个带有权重的新查询，对查询和标记的记录列表进行处理。 如果选择无历史记录的简单搜索模式，则跳过此步骤，直接使用查询频率向量。

最后，计算了数据库的加权查询与TFIDF矩阵之间的余弦相似度。 然后执行快速排序。 快速排序主要是利用分治思想。 也就是说，首先将原问题划分或缩小为较小的子问题，然后递归或迭代求解子问题，最后合成子问题，得到原问题的解。 最后，我们得到了具有文档索引及其相似性分数的排序列表。 然后我们回去，通过索引找到文档并显示它们。

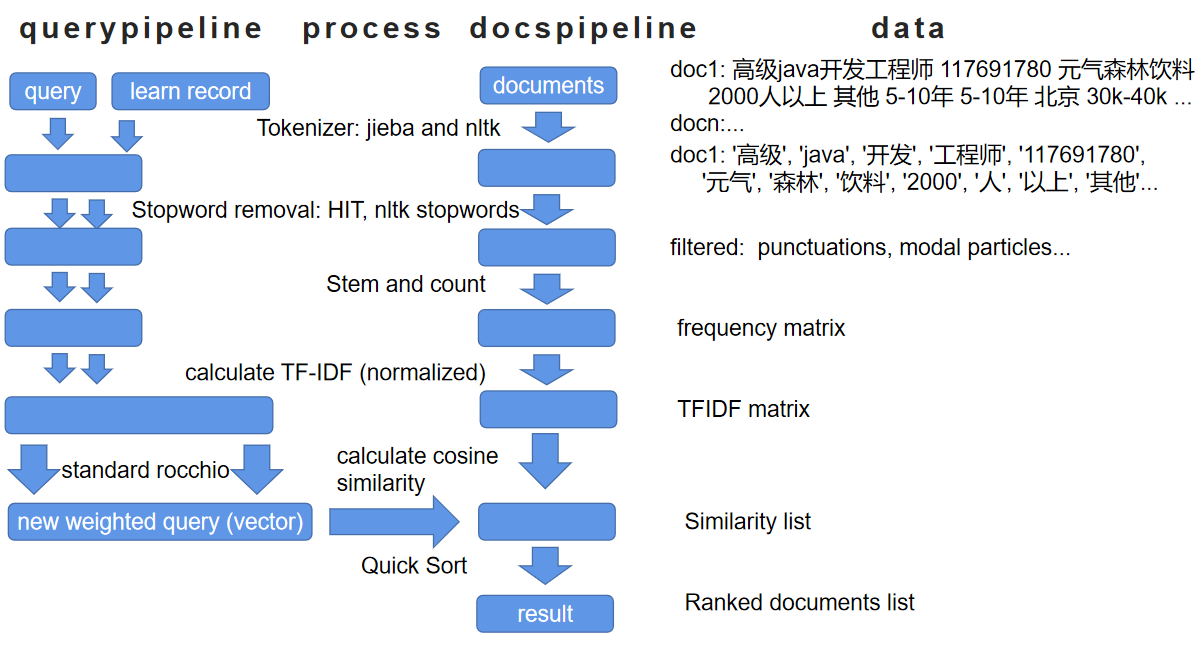


图4。 我们系统的管道

1. 实验和结果

我们在python中使用tkinter来构建我们的用户界面。 我们使用pickle来序列化用户信息。 我们做了一个标准登录系统的完整功能，我们有登录和注册功能，我们有验证，处理任何无效的操作。

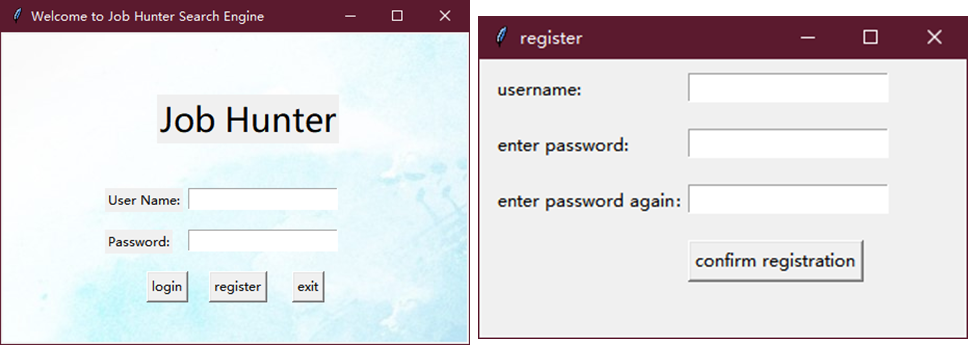


图5.6。 登录窗口

主窗口上方是输入框和按钮，左边是记录用户标记的按钮，右边是显示结果的表格。 在左侧底部，我们还提供了页面快速显示更多的结果。



图7。 简单的搜索与“阿里巴巴杭州2000人以上”

我们的用户可以通过点击左边的按钮将文档标记为相关和无关的，我们的系统可以从记录中学习。 例如，如果我们选择第11条记录作为相关的记录，那么我们使用学习搜索时的结果如图8所示。 请注意，我们的学习搜索没有考虑到当前的输入。



图8。 学习了搜索例子

注意事项：

数字，例如5到10年，小行将被消除，因为停止词和两个数字将成为一起。 所以，我们必须确保数字之间至少有一个空格，我基本上在标记之前用一个空格替换这些空格。

识别信息的重复性可能会影响我们的结果。 例如，每个公司都有一个独特的名称和一个独特的公司ID，所以当我们的系统从标记的文档中学习时，它会从中学到更多的东西，这意味着重复的识别在学习时会有更高的权重，从其他角度来看，同一公司比同一信息更有可能上升排名。

评价：

我们缺乏政府标准。 当我们试图在我们的系统和拉古网站上使用一些示例查询并进行比较以评估我们的结果时，由于我们的数据集相对于整个拉古数据库来说相对较小，它显示出问题，许多相关记录出现在我们的数据库中。 当我们的数据库接近拉古数据库时，我们将来可能会做有意义的评估。 但我们仍然在这里展示了一些评估方法。

设正和负是我们系统分类的相对和无关结果，真和假是我们分类的准确性，这是我们的结果和真实类的比较，同样是正确的，反之亦然。

|  |  |  |
| --- | --- | --- |
|  | 分类正确 | 分类错误 |
| 归类为阳性 | tp | FP |
| 分类为阴性 | 万亿 | fn |

表1：评价基数

一些经典评价：















ROC曲线：Roc曲线是所有可能阈值的FPR和TPR对的曲线，x轴上的FPR和y轴的TPR。 曲线下面积越大，系统越好。

1. 结论和今后的工作

我们的目标是准确快速地获取大量的数据，为后续的分析做准备，比较顺利地完成爬取数据并作为JSON进行分析，创建一个请求类会话，模拟用户向服务器发送请求。 由于对时间间隔的关注，可能拒绝访问的问题也得到了解决。

在实际写作中，发现不能保证在像搜索引擎这样的长词分区的基础上实现完整的对应，这是在安装Jieba库之后解决的。 对于搜索结果，我们发现重复关键词的输入会影响搜索结果，然后通过对TFIDF的分析来解决这个问题。 此外，程序可以通过数组了解用户的选择，从而知道搜索结果是否相关。 我们为数据库存储TFIDF矩阵，解决了冗余计算问题。 此外，程序可以通过数组了解用户的选择，从而知道搜索结果是否相关。

在实验过程中，对于用户登录和文档关联选择，我们的写作比较成功，实验中没有问题。 但发现在处理数字时会出现错误。 所以，我们选择用空格分隔数字。

同时，在实验中我们发现，每次计算时，我们的系统都有点慢。 这是因为我们把查询作为文档，并将它们与文档一起计算，在每次搜索时。 这部分可以从用户系统中分离出来。 为了改进系统，可以预先计算和存储tfidf结果。 然后我们可能只计算标准的rocchio和相似度，这是非常快的。 结果应该马上出现。 再者，重复信息可能会影响权重，虽然没有实际影响，但仍希望在后续工作中得到解决。

我们也希望在这个系统的基础上，我们可以结合用户的搜索内容和查看频率来进行机器学习，从而向用户推荐数据。 这种结合大数据的改进可以保证整个系统更加智能化。

参考资料

1. 马丁，通用。。 （1980）。 求职者的图书馆指南。 职业展望季刊，24，6-11。
2. Bianco，D.，&Maki，K.E。。 （2015）。 求职人的来源：在哪里找到就业线索和其他求职资源。
3. Frakes，W.B.，&Baezayates，R。。 （2004）。 信息检索：数据结构和算法。 普伦提斯大厅，15（5），1211-1214。
4. Rocchio，J.J.。 （1971）。 关联反馈信息检索。 智能检索系统-自动文件处理中的实例。
5. 罗，T，Weck，P.，Sequeira，A.，Tendulkar，N.，Bentov，S.，&Levine，J。。 （2008）。 求职搜索引擎及使用方法。 我们。
6. Silverstein，C.，Marais，H.，Henzinger，M.，&Moricz，M。。 （1999）。 分析一个很大的网页搜索引擎查询日志。 Acm Sigir论坛，33（1），6-12。