PageRank计算

崔轩宁 1800013083

页面信息提取

从英文Wikipedia中下载2021年10月20日的全部页面数据，以此为根据计算页面的PageRank。所下载数据为xml格式，由若干页面组成，以page为单位。为了最终的计算，我们需要提取每一页面的概念、重定向和引用关系，具体细节如下：

概念

在Wikipedia的网页内容中，当前页面表示的概念用"title"属性标识。每一页的概念即为当前页面的"title"值，但并非所有的值都是我们需要的“概念”，有些页面的标题并非是合法的概念，如以"Category: "、"File: "、"Image: "、"Template: "等开头的标题，他们的共性是带有": ", 因此我们采用正则表达式，筛选不含有": "的概念作为符合我们采集条件的概念。我们统计每一个合法的概念，连同其标号用字典存储之。

重定向

在wikipedia的页面中，并不是所有的页面都会被显示出来，有些页面会被重定向到其它页面，用标识符"redirect title="表示。故我们检查每一页中是否有该属性匹配，若有，则该页面没有其它信息，对该页面概念的搜索将会跳转至"redirect title"表示的概念，故我们对该概念不予考虑，这可以通过一个布尔变量实现。

引用关系

我们需要统计每一页对其它页面的引用关系，这在数据文件中是通过"[[]]"实现的。但要注意，并不是所有的"[[]]"都标识一个符合条件的概念引用。正如同“概念”中所提到的，我们只选择不带":"的概念作为合法的概念引用，这同样可以通过正则表达式实现。此外，有些对页面的引用由两个部分组成，中间靠"|"连接。经过与维基页面的对应，我们发现只有前面的部分才是真正有效的页面概念，后面的部分则是该概念所属的种类(Category)，故我们选择前面部分作为引用的概念。考虑到wikipedia总页面数过于庞大，我们采用邻接表而非邻接矩阵的形式加以存储。将所有合法的概念，连同其引用概念组成的列表当作一个字典存储。

综上，我们读取文件，统计上述信息，并将提取的特征保存。这里有一个细节，为了缓解内存压力，同时为了避免因断网等外界因素而导致数据存储失败，我们以一百万条数据为单位分开储存，最终一共各存储了64个package。

信息处理及采样

为了后续PageRank算法的有效进行，我们需要对提取的特征信息进行处理，这主要是指建立若干的映射关系；此外，我们还要从中挑选合适的150w个页面以进行后续的算法计算。

信息处理

我们已经有了概念到编号的映射和概念到引用概念列表的映射。考虑到我们的操作是在编号上进行，最后的输出却需要概念的名称，故我们还需要建立：

- 编号到概念的映射(这种映射是一一对应的)
- 概念编号到引用概念编号列表的映射

采样

我们一开始采用对所有概念随机sample的方式，但很快我们便淘汰了这种方法，因为这样搜集的页面往往不具有太多逻辑的相关性，彼此之间的引用较少，计算得到的结果可解释性差。为此，我们采取一种新的采样方法：

我们首先随机选择一个概念，再选择这个概念的某一个引用概念，不断依次递归下去，最终采样到符合条件的页面数。

这样做的好处是所选择的概念中具有一定的相关性，最终计算出的结果可解释性强。在采样完成后，我们依次检查每个采样出的概念的引用概念，如若该引用概念不在·我们采样的概念列表中，则将其剔除，这样做减少泄漏，更有利于得出有说服力的PageRank值。

综上，我们采样出了待研究的页面，并准备好了PageRank算法所需要的条件。此外，为了内存的合理利用，我们并未存取信息处理中提到的两个映射，而是只存取了其中被采样出的部分。

PageRank算法

本节先来介绍一下PageRank算法的思想，然后介绍我们的实现方式及实验结果

算法思想

PageRank算法是计算互联网网页重要度的经典算法，它对每个网页给出一个正实数，表示网页的重要程度，整体构成一个向量，PageRank值越高，网页就越重要，在互联网搜索的排序中可能就被排在前面。

算法的基本思想是在有向图上定义一个随机游走模型，即一阶马尔可夫链，描述随机游走者沿着有向图随机访问各个结点的行为。在一定条件下，极限情况访问每个结点的概率收敛到平稳分布，这时各个结点的平稳概率值就是其PageRank值，表示结点的重要度。

在实际的有向图中，考虑一个在该图上随机游走的一阶马尔可夫链，假定其从一个结点到其连出的所有结点的转移概率相等，那么这个马尔可夫链未必具有平稳分布。为了解决这个问题，我们考虑一个完全随机的转移矩阵，即从任意一个结点到任意一个结点的转移概率都是1/n，将这个转移矩阵和原转移矩阵线性组合成新的转移矩阵。可以证明，这个一般随机游走的马尔可夫链存在平稳分布，我们假设线性组合系数为阻尼因子d，则PageRank值(我们用R表示)可以由如下公式决定：

$$R = dMR + \frac{1-d}{n}1$$

实现方式

显然，我们可以采用迭代法来得到R的近似平稳分布。在实际计算中，我们取阻尼因子d为经验值0.85，初始PageRank值设为1，相邻两次迭代误差采用L2范式计算，阈值eps设为0.001。最终经过测试，在68次迭代后，R趋向于收敛，L2范式误差为0.00088，小于阈值eps。注意，由于我们将PageRank的初始值设为1，因为我们前面排除了泄漏的情况，故PageRank总和为n，故完全随机的转移矩阵带来的最终结果是每个点有1-d的PageRank值，这在代码中可以简化我们的实现，同时提升计算的精度。

实验结果

我们统计了排名前20的数据，列举如下：

1	United States	6	France	11	United States Census Bureau	16	Canada
2	The New York Times	7	London	12	Italy	17	Wikidata
3	World War II	8	The Guardian	13	English language	18	World War I
4	United Kingdom	9	India	14	Australia	19	Japan
5	New York City	10	Germany	15	England	20	Democratic Party (United States)

PageRank结果分析

具有一定代表性

我们观察前20名所表示的概念，发现都是我们耳熟能详的概念，这些概念在我们的生活中有较大影响，其重要性较高，这说明PageRank所统计的排名在一定程度上确能反映网页的重要程度

从属关系起到影响

可以看到，一般来说，比较大的概念相对于比较小的概念往往更为重要，前20名出现了很多国家，极少数的大城市，完全没有小城市，这刚好符合国家>大都市>小城市的重要情况

概念的延伸含义

我们可以看到，纽约作为一个城市，排在所有采样出概念的第五名。如果只是从城市本身的角度来看，尽管纽约是人尽皆知的大都市，也可能不会有如此高的排名。更多的可能还是纽约所具有的抽象意义，如纽约可以联想到都市生活、大型企业，这些都使得它有大量的链接引入，从而具有更高的网页重要性

相似概念具有相似的PageRank值

如排在第二十位的是美国的民主党，而共和党也是排在第三十位，这在一百万的采样中距离是十分相近的，说明二者的重要程度相近，这与二者在现实世界中的重要性几乎是一致的。还有很多其他的概念属性也是如此