人物信息检索系统的功能及其实现

周聿浩
2016011347

September 14, 2017

1 简介

这是一个利用 Django 搭建的一个人物信息检索系统，大约从 Wikipedia 爬取了 10000 个人物信息，并且提取了其中 Infobox 的对应信息。

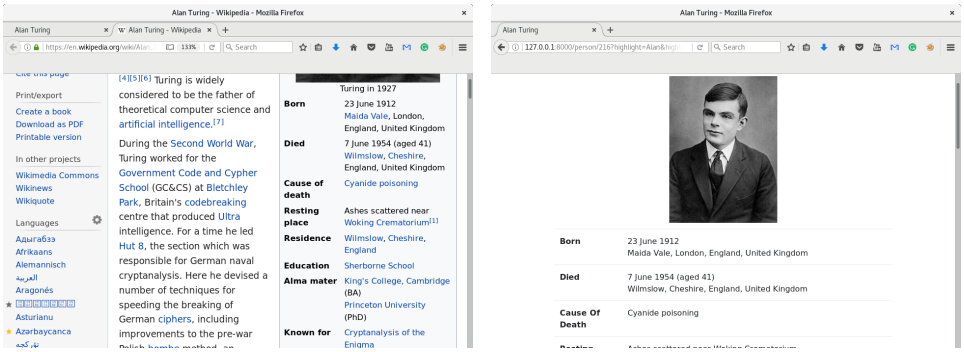

Figure 1: 对于 Wikipedia 中爬取的信息，我们重新组织了其格式并且进行显示。

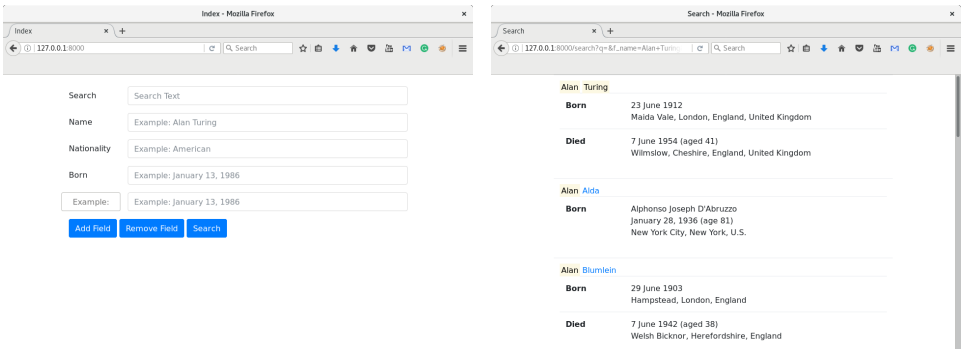

Figure 2: 左侧为搜索页面，右侧为搜索结果，匹配的字段被高亮显示。

对于已经爬取的信息，我们提供了一个对其进行搜索的页面，可以根据关键词在其中搜索，并且还可以根据原先 Infobox 中的标题进行特定字段的查询（例如 Born、Died、Name、Nationality 等），同时还可以让用户自行添加可以查询的字段。

搜索的结果按照匹配的关键字个数从高到底排序后显示，如果结果过多将会分页显示。同时匹配的关键字会被高亮标出。

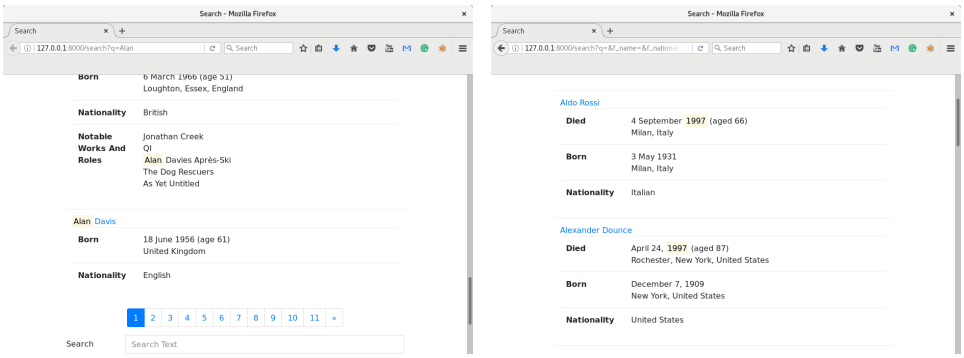

Figure 3: 左侧为搜索结果过多时的分页显示效果，右侧为按照字段搜索 Born 中含 1997 的人物结果。

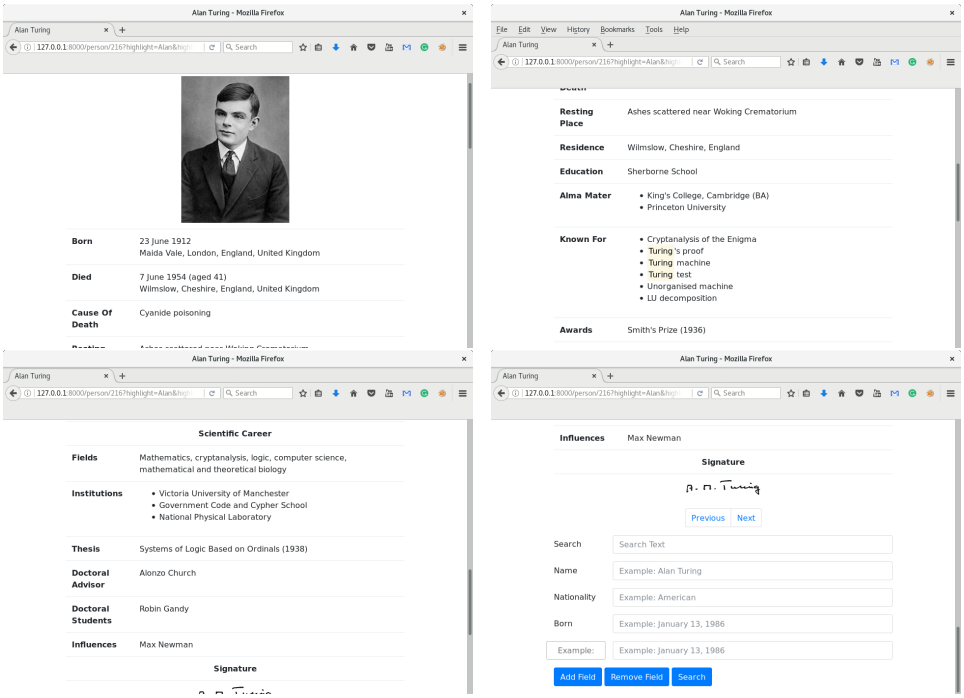

Figure 4: Alan Turing 信息的展现。

2 部分实现细节

爬虫部分利用 BeautifulSoup 来处理获取的页面，提取 Infobox 中的信息。

具体来说，人物超链接的爬取是通过寻找 ID 为 *mw-content-text* 的元素下所有 *li* 标签的第一个超链接来实现的。在爬取完毕后检查是否存在 infobox，如果存在则开始提取信息。由于其中信息具有一定规律（例如大部分信息是以标题、内容的形式来组织的），只需要用 BeautifulSoup 提取相应的 *<th></th>* 以及 *<td></td>* 部分即可。

前端界面利用 Bootstrap 来优化显示效果。

关于数据的存储，在提取出信息后利用 JSON 来保存在 sqlite 数据库中，并且额外提取出一个关键字字符串用于搜索。对于每个人物都会分配一个唯一的 ID 以方便索引。

分页功能利用了 Django 自带的 Paginator 类。查询关键词的高亮以及自定义字段搜索框的增加与删除使用 Javascript 在前端完成。