## 浅谈算法：关于智能爬虫的探索

## 引子

如果当初让我去问顾客他们想要什么，他们肯定会告诉我：“一匹更快的马。” -- 亨利·福特

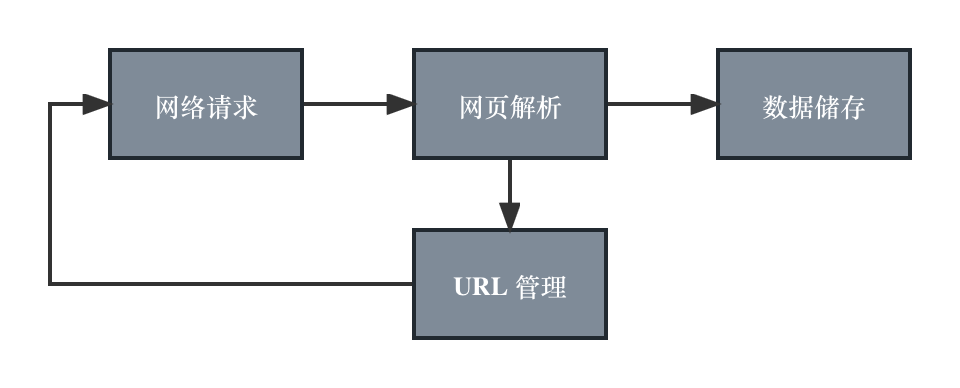
如今是人工智能大红大紫的时代，不管是 ChatGPT 还是随之而来的各种智能应用，让许多人看到即将到来的在几年前几乎不可想象的科幻世界。不过，在爬虫领域，人工智能似乎还没有过多涉及。诚然，爬虫作为一项“古老”的技术，在过去 20 多年缔造了搜索引擎、新闻聚合、数据分析等多而又多的技术行业，不过在如今还暂时没有看到明显的技术突破：爬虫工程师还是主要依赖于 XPath、逆向等技术来自动获取网络数据。不过，随着人工智能和机器学习的发展，爬虫技术理论上也可以做到“自动驾驶”。本篇文章，将从多个维度介绍所谓**智能爬虫**（智能化、自动化数据提取爬虫技术）的现状以及可能的未来发展方向。

## 爬虫技术现状

爬虫程序是一种自动化的程序，用于从互联网或其他计算机网络上获取数据。它们通常使用自动化数据抓取技术来自动访问网站，并收集、解析和存储网站上的信息。这些信息可以是结构化或非结构化数据。

传统意义上的爬虫技术主要包括以下模块或系统：

1. **网络请求**：对网站或网页发起 HTTP 请求，获取 HTML 等数据；
2. **网页解析**：对 HTML 进行解析，形成结构化的树形结构，并通过 XPath 或 CSS Selector 获取目标数据；
3. **数据储存**：将解析后的结构化数据进行存储，可以是数据库，也可以是文件形式；
4. **URL 管理**：负责管理待抓取的URL列表和已经抓取的URL列表，例如分页或列表页的 URL 解析和请求。



以上都是基础的爬虫技术模块，对于大型爬虫系统来说，还需要具备任务调度、错误管理、日志管理等生产环境必要模块。笔者的爬虫管理平台 [Crawlab](https://www.crawlab.cn) 就是用于企业级生产环境爬虫管理平台。另外，针对一些反爬措施，例如验证码或 IP 封锁，通常还需要额外模块，例如验证码识别、IP 代理等。

然而，目前爬虫程序的开发主要集中在网页解析上，这也是非常消耗人力的地方。虽然 HTML 需要解析的网页数据，但是不同网站的网页布局、排版、风格、内容各不相同，导致每个网站、网页都需要单独编写解析逻辑，从而大大增加了人工编写成本。虽然一些通用爬虫例如搜索引擎爬虫可以不用编写过多解析逻辑，但这样的爬虫通常都无法聚焦于某些特定的主题数据提取。因此，为了降低人工编写成本，最好是不需要编写或少量编写解析逻辑就可以自动提取网页数据，这也就是智能爬虫的主要目标。

## 已知实现方式

要实现智能化网页提取并不容易，但目前已经有一些对于智能爬虫开发的尝试。其中，青南开发的 [GNE](https://github.com/GeneralNewsExtractor/GeneralNewsExtractor) (GeneralNewsExtractor) 是网页正文提取的开源实现，基于[文本及标点符号密度提取算法](https://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CJFQ&dbname=CJFDLAST2019&filename=GWDZ201908029&v=MDY4MTRxVHJXTTFGckNVUkxPZmJ1Wm5GQ2poVXJyQklqclBkTEc0SDlqTXA0OUhiWVI4ZVgxTHV4WVM3RGgxVDM=)。崔庆才开发的 [GerapyAutoExtractor](https://github.com/Gerapy/GerapyAutoExtractor) 基于[列表簇及 SVM 算法](https://cuiqingcai.com/9531.html) 实现了网页列表页识别。[八爪鱼](https://www.octoparse.com/) (Octoparse)，一个商业客户端软件，已经开发出自动列表识别模块。国外的 [Diffbot](https://www.diffbot.com/) 是一个基于 API 的智能网页识别平台，识别准确率非常高，号称有 99%。对于已知的智能爬虫实现方式，目前主要基于网页 HTML 结构以及内容，例如 GNE 和 GerapyAutoExtractor。对于商用软件例如八爪鱼、Diffbot，我们无法得知具体实现方式。

## 探索列表页识别

现在正文识别的准确率已经非常高了，具体也有很多技术实现和应用。在这里我们主要关注列表页的识别，这是很多爬虫程序的网页解析工作。

我们可以从经验来推断如何自动识别想要的内容。人类是视觉动物。当我们看到一个文章列表的网页，不出意外会立马识别到这个文章的列表，如下图。但是，我们究竟是如何识别出来的呢？其实，我们自然将同一类的文章列表项归为一类了。因此，我们会很快意识到这其实是一个列表页。当然，为何这些列表项是相似的呢？我们可以看到这些列表中的子元素也相近，因此自然而然将其联系在一起。各个子元素加总在一起成为单独一个列表项，然后我们的大脑自动将其归为一类。这就是列表页识别的过程。



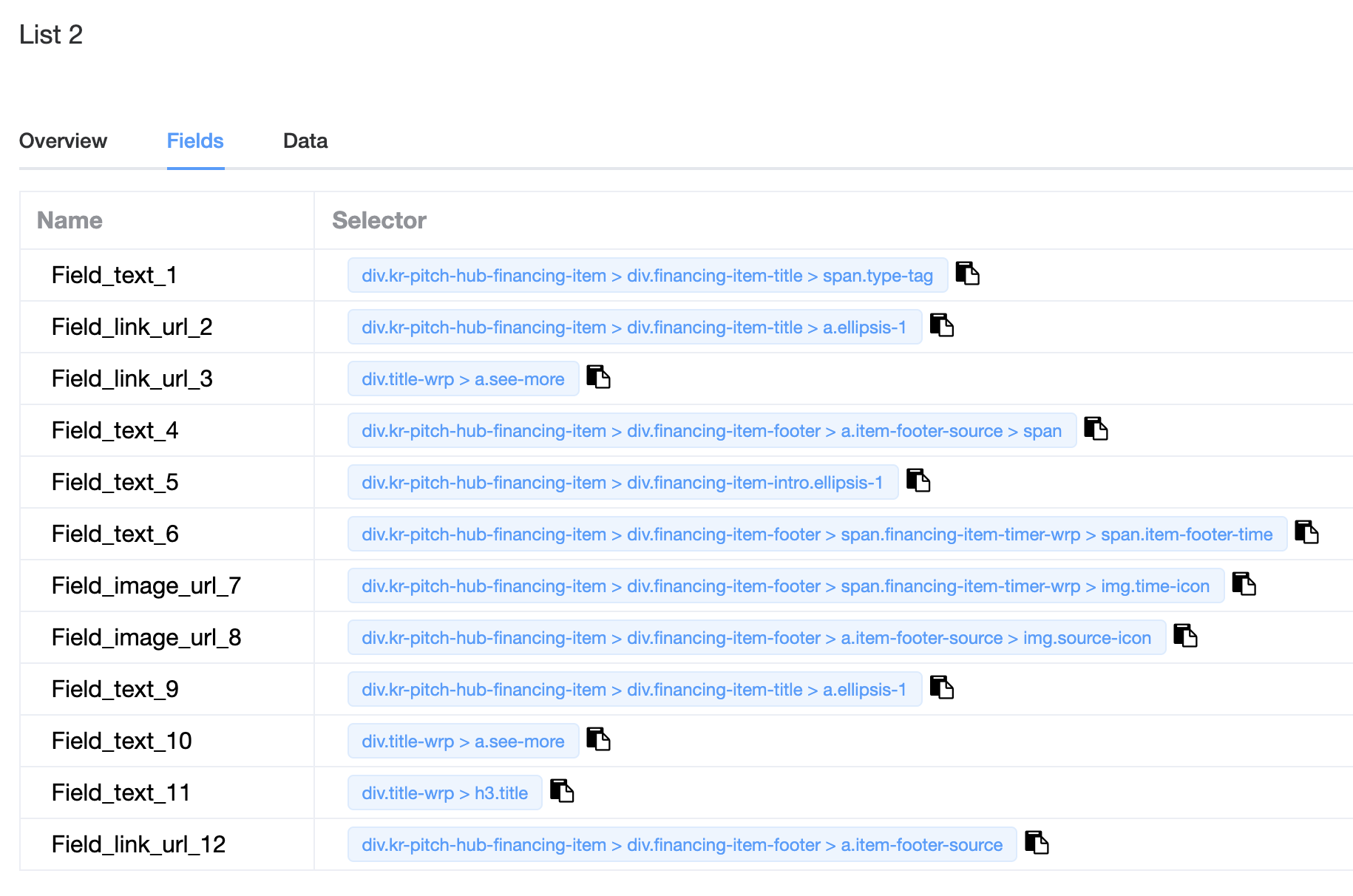
基于这样的分析，其实非常容易联想到机器学习中的聚类算法，我们需要做的只是将网页各个节点特征提取出来，然后利用聚类算法筛选出同样类别的节点即可。当然，这里的特征选择需要重点考虑，我们不是单纯的将 HTML 的单个节点取出来看，而是需要将其与其他节点一起关联起来提取特征，这样的话就可以得到一些不同类别的节点。然后，我们可以根据节点簇的整体信息来筛选出想要的列表页。

当然，要用代码实际实现这样的算法，并不是一件容易的事情，需要将 HTML 各个节点进行建模（Modelling）、向量化（Vectorization）、以及基于此构建树状图（Tree-like Graph）。这是非常繁琐的事情。不过所幸的是，笔者已经利用 sklearn、networkx 等库实现了一个基础的列表页识别系统 [Webspot](https://github.com/crawlab-team/webspot)，可以自动识别一个列表页上的列表元素，而且可以视觉化展示识别的结果，如下图。



对于大部分列表来说，Webspot 的识别能力还是不错的。虽然达不到 Diffbot 的准确度，但目前来说对于不是非常复杂的网页还是能够准确识别出来的。

那么，既然已经有 Diffbot 这样的列表页识别方案，为什么还要新造轮子呢？其中一个最重要的原因是，商用且准确率高的软件例如 Diffbot 不能直接给出 XPath、CSS Selector 这样的可以重复使用的提取规则，而我们自动识别其实只需要拿到需要的提取规则即可，然后通过集成到 Scrapy、Colly 这样的开源爬虫程序里，就可以大大节省数据抓取的成本。这也是 Webspot 目前能够带给用户的一个功能，不仅能识别到列表页元素以及相应字段，还能够给出提取规则，如下图。



有了这样的提取规则，只需要自动识别一次，就可以在相似的网页中自动提取数据了。

当前，Webspot 还处于开发早期阶段，之后应该会有更多新功能以及算法的开发和优化。

## 未来发展

智能爬虫相当于网页上的自动驾驶，不需要太多人工的操作就可以让爬虫按照要求获取想要的数据或信息，这是很多数据需求者、爬虫工程师的理想技术。但是，目前智能爬虫还不太成熟，已有的实现和技术相对较为简单，未来或许可以通过深度学习、强化学习等技术来提高智能爬虫的识别能力。另外，图论与人工智能的结合，配合视觉技术或许可以让智能爬虫在准确率方面有所突破。笔者会通过 Webspot 这个项目继续在智能爬虫上进行探索，解决数据提取的成本问题。对智能爬虫发展感兴趣的朋友可以加笔者创建的交流群。



## 社区

如果您对笔者的文章感兴趣，可以加笔者微信 tikazyq1 并注明 "码之道"，笔者会将你拉入 "码之道" 交流群。