

北邮人 BT 自动筛种做种系统

1、系统概述

1.1、系统简介

北邮人 BT 作为国内高校间首屈一指的教育网 BT 系统，一直以来受到了各高校师生的一致好评。目前，北邮人 BT 实际注册人数已经超过 31 万人，系统内存有涵盖各类电影、剧集、动漫、音乐、游戏、资料、体育和纪录片等各类种子六万三千多个，总下载量达到 82.911 PB，总数据量达到 335.213 PB。北邮人 BT 已经成为了北邮及其他高校同学学习、科研和生活的必备网站。

基于基本的 BT 秩序，即“人人为我，我为人人”，北邮人 BT 也和六维空间、蒲公英等其他 BT 系统一样，需要计算每位用户的分享率，当分享率低于一定数值账号会被封禁而无法使用。为了在下载资源的同时不会被封号，用户必须上传资源给其他用户下载，或者将下载的资源进行做种（提供给其他用户进行 P2P 下载）。通常情况下，大多数用户没有合法的资源直接上传（北邮人 BT 拥有严格的种子发布策略），因此只能选择做种提高分享率。

本系统主要帮助使用者进行自动筛选种子并做种，从而累计分享率，避免现有模式下用户需要频繁手动操作的痛点。本系统只需要用户填写用户名和密码，运行后自动完成后续任务，使用简单方便。具体来说，本系统首先进行模拟登录，使用用户的账号和密码，并通过验证码自识别进行登录，然后使用爬虫技术快速列出当前的热门种子，按照用户自定义的一些筛选策略和系统自身的算法进行选择，然后下载该组种子，最后，系统将自动将种子添加到开源的 bt 下载器中进行下载，下载完成后自动做种。

本系统的定位为：基于客户端的插件式小工具，采用独立安装部署的方式，直接在用户的个人电脑中运行即可，无需中心服务器，具有安装快速，功能强健，简单易用，可长期使用等优点。

下面对本文中的一些术语进行简要解释：

- **BT (BitTorrent):** BT 是一个文件分发协议，它通过 URL 识别内容并且和网络无缝结合。是一种多点共享协议软件，由美国加州一名叫 Bram Cohen 的程序员开发出来。它对比 HTTP/FTP 协议, MMS/RTSP 流媒体协议等下载方式的优势在于，一个文件的下载者们下载的同时也在不断互相上传数据，使文件资源可以在不同用户之间传输。
- **Tracker:** 提供 BT 的服务器称为 Tracker，记录了所有资源和用户的相关信息，如上传量、下载量和分享率等。Tracker 实际上就是一个网址，拥有特定的域名和 URL。
- **种子文件:** 以扩展名 .torrent 为后缀的文件，包含资源的创建时间、制作者填写的注释、以及制作者的信息等资源的必要描述信息，以及资源块的校验信息（资源文件会按照固定大小分块，不同块可以从不同用户得到）。此外，种子文件中还包含 Tracker 服务器的地址以及用户的标识信息如北邮人 bt 中的 passkey。种子文件通常很小，大约几十 K、几百 K 大小。
- **BT 的运行机制:** 由集中式的服务器记录关键信息，所有的资源保存在用户本地电脑，用户通过集中式的服务器找到拥有资源的用户，分别建立 P2P 连接进行下载。具体来说，每个资源会按照一定大小分块，不同块可以从不同用户得到，用户可以多线程从多个用户下载从而满带宽进行数据传输，提高下载速率。简单来说，BT

是一种“我为人人，人人为我”的下载方式。

- BT 的维护机制：用户具有一些基本属性，包括分享率、下载量、上传量和魔力值，当用户下载时按照下载数据量进行记录，同理可得上传量。 $\text{分享率} = \text{上传}/\text{下载}$ 。分享率高的用户在站点拥有更高的权限如发布资源种子等。当分享率持续低于一定值时将被封禁账号。魔力值等同于积分，可以按照一定比例兑换上传量或其他礼品。
- BT 的技术细节：用户通过网站管理账户信息，搜索资源，当用户希望下载该资源如一部电影时，用户从网站上下载种子文件到个人计算机，由于种子很小，下载时间将会很短。种子需要使用一些特定的 BT 客户端，通过解析种子找到 Tracker，并向其发送用户的 passkey，Tracker 校验后如果合法则返回拥有当前资源的所有做种用户，客户端得到用户的地址列表，分别和每个用户建立 P2P 连接，请求特定编号的资源块，最终凑齐全部资源块（具体细节例如随机扰动等不做过多介绍），通过种子文件进行校验，如果无误则下载完成。下载完成后自动开始做种，向服务器定时发送心跳包表明在线，如果无人下载依然会被奖励魔力值。

1.2、系统目标

基于本系统的定位及功能要求，本系统要实现的目标主要是：北邮人 BT 的自动筛种和做种系统。具体而言，需要实现如下功能目标：（依照使用流程顺序排列）

- 1) 系统需要能够从 BT 站点获取到当前热门种子的详细信息；
- 2) 系统必须能够对种子的详细信息进行分析处理，并根据筛选策略通过算法找出最优的一个或多个种子；
- 3) 系统必须能够自动从服务器中下载种子文件到本地；
- 4) 系统需要根据种子文件自动开始下载，且当下载完成后自动做种；
- 5) 基于 1)、3)和 4)，系统必须向服务器证实自己的身份，即模拟登录，从而获取必要的种子信息，且能够合法下载到资源，且在做种时能够将行为记录以提高用户的分享率；
- 6) 基于 2)和 4)，系统需要支持用户自动配置筛选策略和一些必要配置，包括但不限于设置做种个数或区间范围、设置做种收益阈值、设置最大占用磁盘空间大小或区间范围以及系统运行时间或做种周期等；
- 7) 基于 5)，系统必须能够自动识别并填充验证码，从而模拟登陆；
- 8) 为了更高的用户体验，系统必须具有美观的图形化交互界面，用于和用户进行交互，如填入用户名和密码、进行必要的设置等；
- 9) 为了更流畅的用户交互，系统需要有相应的进度展示、错误提醒、版本适用情况说明和 log 日志。

此外，本系统还需具有以下性能目标：

- 1) 系统的用户交互需要具有即时响应，如需等待则给出必要提示；
- 2) 除非网络因素影响，系统的模拟登录需要在用户可接受范围内，一般为 1s 内；
- 3) 除非网络因素影响，系统的资源下载和做种速率须与其他手动操作下载器相同；
- 4) 系统的信息处理（主要指种子筛选）等待时间需要在用户可接受范围内，一般不超过 2~3 分钟；
- 5) 本软件运行后占用的系统内存不超过 40MB，不计算种子和资源的空间占用，本软件占用的磁盘空间不超过 80MB；
- 6) 软件运行时对 CPU 的占用、磁盘的读写和网络的收发应在合理范围内；
- 7) 考虑到软件需要长期运行（通常为几周到几月不等），软件应具有合理的磁盘空间

和内存维护，严格避免如操作指针等带来的内存泄漏。

1.3、系统运行环境

硬件平台：个人电脑

操作系统：Linux，具体为 Ubuntu 16.04 TSL，安装 Python3.x 运行环境，需要支持 bash

数据库系统：无要求

编程平台：PyCharm 2019.1.3 (Professional Edition)、Qt4.8.2

网络协议：需要具有 IPV6 地址以成功访问 BT 服务器和 Tracker，做种使用 UDP 进行 P2P 连接，与 BT 服务器通信建立 TCP 连接，使用 HTTPS 协议，用于模拟登录、信息爬取和种子下载。

1.4、开发环境

主要开发语言：python 3.7(装有 opencv-python4.1.1.26)、bash(GNU bash, version 4.1.2(1)-release(x86_64-redhat-linux-gnu)) 和 C++ 2013 (Professional Edition)

开发工具：PyCharm 2019.1.3 (Professional Edition)、Qt4.8.2、Ubuntu 16.04 TSL

2、总体结构设计

2.1、软件结构

根据软件的功能需求定位，按照高内聚低耦合的原则划分为以下模块：模拟登陆模块、验证码识别模块、爬虫模块、transmission 控制模块、定时更新做种列表模块、逻辑控制模块、种子评分模块以及界面展示模块。模块间的关系如图 2-1 所示。

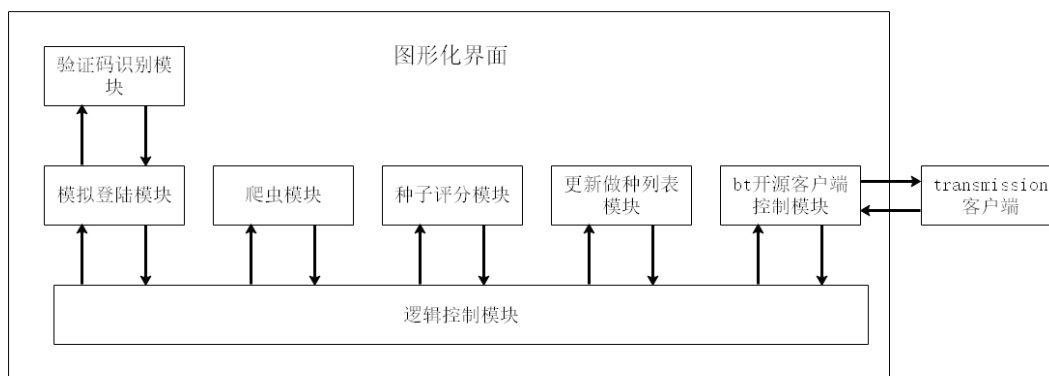


图 2-1 模块划分及其关系图

此外，系统的工作流程图如图 2-2 所示。

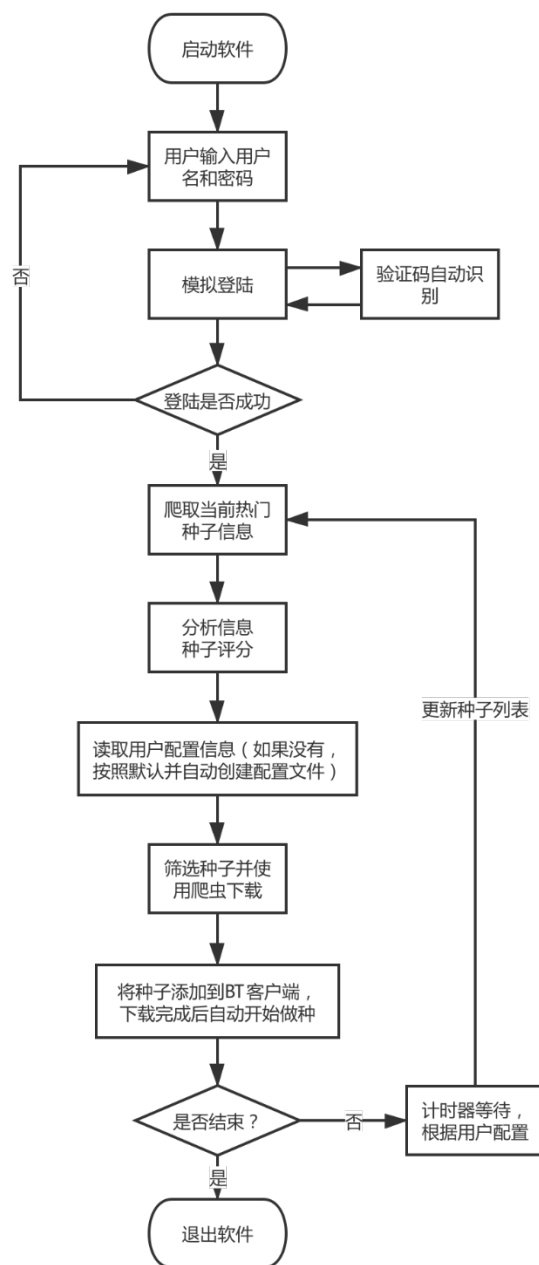


图 2-2 工作流程图

2.2、设计思想

根据软件的核心功能,软件需要获取种子并进行下载,由于已经有现有开源客户端软件,我们尝试对其进行调用,因此封装单独模块来进行操纵。若需要获取种子需要登录北邮 BT 网站进行下载,因此设计模拟登录模块,其中登录涉及到验证码破解,封装单独模块进行处理。需要获取所有种子信息,因此设计爬虫模块。在处理种子信息时,需要使用一定的算法策略,设计种子评分模块。客户端长时间运行,种子需要定时刷新,设计更新模块。此外全部运行流程需要逻辑控制模块作为统一入口,同时通过命令行进行调用。与用户的交互操作由图形化界面完成。

3、模块设计

3.1、模拟登陆模块

3.1.1、功能描述

该模块完成获取 login 界面的必填信息，将登录信息提交至 takelogin 界面完成登陆，获取登陆后的 cookies 以备之后请求信息和下载文件。

3.1.2、接口描述

模拟登陆模块的代码名称为 login.py。

登陆所需的图片验证码信息从 VCRmodule 模块中获得，VCRmodule 模块的接口是 imgstring，表示已破译的图片验证码。

Login 模块的输入信息包括固定不变的请求头 header，其中含有 user-agent 信息；login 界面 URL；提交登录信息的 takelogin 界面 URL；登陆用户名、密码、图片 hash 值、图片验证码。

Login 模块的输出信息为 session.cookies，是登陆成功的会话的 cookie，留做爬虫模块请求网页使用。

3.1.3、数据结构描述

本模块采用顺序存储，无特殊数据结构。

3.1.4、实现思路

给出实现该模块的基本思路，包括对模块内部结构、算法、编程方法等的初步设想。

本模块的主函数完成以下三个功能：①向 login 界面请求获得验证码图片；②向 takelogin 界面提交登录信息，获取返回 cookie 值；③使用 cookie 值获取目标界面网页信息。

功能①使用 getimglink 函数从 login 网页信息中获取图片下载链接，使用 downloading 函数对下载链接进行请求，完成图片的下载。

功能②的登录信息由四部分组成：用户名、用户密码、图片 HASH 值、图片验证码。其中，用户名、用户密码是我们手动输入的；经调试，发现图片 HASH 值是图片下载链接的一部分，采用 char_set 函数对图片下载链接进行 split 操作，返回 HASH 值；至于图片验证码，我们将所下载的验证码图片的文件路径提供给 VCRmodule 模块里的 main 函数，由 main 函数输出破解好的验证码。

功能③继承功能②中从 takelogin 界面返回的 cookie 值，建立网页请求会话，并保留会话的 cookie 值。

3.1.5、其他

选写另外一些有必要叙述的部分，如模块的表现形式（以何种形式运行，如服务程序、应用程序、数据库脚本等等）、有无人机交互界面（有则给出其类型和设计原则）等。

3.2、验证码识别模块

3.2.1、功能描述

对登录验证码进行识别，返回识别结果字符串。

3.2.2、接口描述

本模块由登录接口调用，输入信息为验证码图片地址，输出信息为验证码识别结果，即六位字母（大写）与数字混合字符的识别结果，执行异常后返回错误信息。

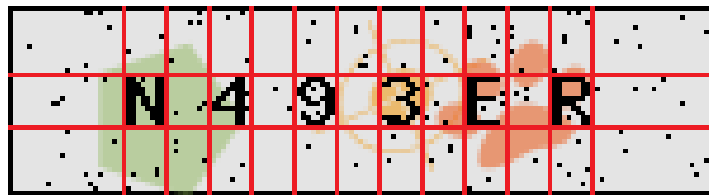
3.2.3、数据结构描述

本模块主要使用的数据结构为二维数组，即验证码图片在代码中的表现形式，此外还有

颜色值，即（R，G，B）形式的数据格式表示当前像素点。

3.2.4、实现思路

首先爬取了一万条验证码图片到本地进行分析比对，试图找到验证码识别方法，初步拟采用调用第三方 API 方式。后来通过观察，发现了一些验证码的规律。



验证码主要由六位字符构成，字符有字母且均为大写字母和阿拉伯数字构成，通知字符位置固定，字符写法固定（即特定字符占有的像素点固定），字符颜色为黑色。因此，处理方法是①建立字符标准模板库，包含了全部可出现的字符；②读取图片并进行切分，分离出待比较的六个字符所在位置的全部方框，方框位置通过经验取得；③对单一字符图片进行二值处理，此时可观察到图像中只存在黑色噪点；④将字符图像与标准字符模板进行逐一比较，具体则是对相应相对位置的像素点做差，可以得出与全部字符的 error 点数目，可认为该数目最小的字符即为最终的识别结果；⑤查找匹配结果对应的字符得到全部六个字符后组成字符串返回。

3.2.5、其他

模块的表现形式为 python 文件，使用 python 语言进行编写，需要调用 cv 库，以函数调用方式被上层函数调用，不创建线程，一次性调用，无人机交互界面，含有常规 IO 数据流和 cv 数据流，须读取本地文件，不写入任何数据。

经过实际测试发现，验证码识别准确率能够达到 99.85%以上，满足实际使用要求，如果识别错误影响不大，重新输入用户名和密码登录即可。

3.3、爬虫模块

3.3.1、功能描述

该模块完成爬取热门种子的各类信息，并下载其种子文件的功能。

3.3.2、接口描述

爬虫模块的代码名称为 hit.py。

此模块继承 login 模块的 session.cookie 值，用来实现本模块中各网页信息的获取。

该模块的输入信息为热门种子界面 URL、欲获取的种子数目、种子文件的存储路径。

该模块的输出信息放在名为 torrent_information 的 csv 文件里，除此之外还有下载好的种子文件。

3.3.3、数据结构描述

本模块采用顺序存储，无特殊数据结构。

3.3.4、实现思路

本模块的主函数完成以下三个功能：①爬取种子的各类信息；②完成种子文件的下载；③将种子信息写入 csv 文件。

功能①中的各类信息包括：排名、资源名称、资源详情页链接、种子名称、下载类型、评论数、生存时间、资源大小、做种数、下载数、完成数、发布者。这些信息都从热门种子界面的 HTML 中获得，各自用一个函数获取，函数返回值为该类信息。

功能②用 intro 函数进入资源详情页，获取种子下载链接，用 download 函数请求该链接，将返回数据写入文件中，完成种子文件的下载。

功能③建立 csvfile，以 excel 格式写入以上信息。

3.3.5、其他

选写另外一些有必要叙述的部分，如模块的表现形式（以何种形式运行，如服务程序、应用程序、数据库脚本等等）、有无人机交互界面（有则给出其类型和设计原则）等。

3.4、transmission 控制模块

3.4.1、功能描述

该模块读取.conf 文件中对做种软件 transmission 的控制信息，并传递到 transmission 的配置文件中，实现对 transmission 的控制与调用。

3.4.2、接口描述

transmission 控制模块的代码名称为 initialize.py。

所需要传递的配置信息从 bt_auto.conf 文件中读出。该文件为脚本文件，供控制台直接调用，因此并无接口。

3.4.3、数据结构描述

本功能区域使用了散列表存储数据。散列表（Hash table，也叫哈希表），是根据关键码值(Key value)而直接进行访问的数据结构。也就是说，它通过把关键码值映射到表中一个位置来访问记录，以加快查找的速度。这个映射函数叫做散列函数，存放记录的数组叫做散列表。

3.4.4、实现思路

给出实现该模块的基本思路，包括对模块内部结构、算法、编程方法等的初步设想。

本模块主要完成以下功能：从.conf 文件中读取对做种软件的控制信息。实现思路是根据从.conf 中读出的控制信息，修改做种软件 transmission 的配置信息 settings.json，从而实现对 transmission 的间接控制。

3.4.5、其他

选写另外一些有必要叙述的部分，如模块的表现形式（以何种形式运行，如服务程序、应用程序、数据库脚本等等）、有无人机交互界面（有则给出其类型和设计原则）等。

3.5、定时更新做种列表模块

3.5.1、功能描述

该模块读取.conf 文件中对定时更新功能的控制信息，进而控制做种列表的更新与维护。每当更新计时结束，首先通过爬虫模块更新种子库信息，再根据种子评价模块从高到低对种子库中的种子进行评分。取前 n（n 从.conf 文件中读出，受用户控制）个种子进行下载与做种，完成该次更新，并进行新一轮的计时。其中如果程序中途退出，计时信息也会得到保存，在下次程序开启时作为起始时间赋值给计时器。

3.5.2、接口描述

定时更新做种列表模块的代码名称为 timer.py。

该模块接受的外部接口为 hit.main 函数、evaluate.main 函数、以及 bt_auton.conf 与 transmission_settings.json 的控制信息。该模块为逻辑控制模块提供了 main 函数接口。逻辑控制模块通过对 main 函数的调用实现前述的相关功能。

3.5.3、数据结构描述

本模块采用顺序存储，无特殊数据结构。

3.5.4、实现思路

给出实现该模块的基本思路，包括对模块内部结构、算法、编程方法等的初步设想。

本模块中所涉及的具有控制功能的局部变量同样要从 bt_auto.conf 文件中读出。使用 for

循环与 `time.sleep` 函数对时间进行计时，每当计时完成时调用 `hit.mian()` 函数进行种子库更新操作。更新完种子库后，调用 `evaluate.mian()` 函数对种子库文件进行评分，根据分数由高到低顺序对种子库中的种子进行排序，取前 `n` 个种子放入做种软件 `transmission` 的监视文件夹中进行下载与做种。上述表述为该模块的一个完整运行过程与实现思路。

3.5.5、其他

选写另外一些有必要叙述的部分，如模块的表现形式（以何种形式运行，如服务程序、应用程序、数据库脚本等等）、有无人机交互界面（有则给出其类型和设计原则）等。

3.6、逻辑控制模块

3.6.1、功能描述

该模块为 `bash` 脚本，用以实现各个 `python` 脚本的信息传递与调用。

3.6.2、接口描述

`Close.sh` 脚本使做种程序正常关闭，`work.sh` 脚本开启做种程序。

`Close.sh` 脚本通过对 `stop_label` 变量的修改来正常关闭由 `work.sh` 控制的做种程序。`Work.sh` 则是使用各个模块以及做种软件 `transmission` 提供的接口开启与维持做种程序的运行。

3.6.3、数据结构描述

本模块采用顺序存储，无特殊数据结构。

3.6.4、实现思路

给出实现该模块的基本思路，包括对模块内部结构、算法、编程方法等的初步设想。

`Close.sh` 脚本通过对 `stop_label` 变量的修改来正常关闭由 `work.sh` 控制的做种程序。`Work.sh` 则是首先正常关闭做种软件 `transmission`，然后执行 `initialize.py` 脚本初始化控制信息，并通过 `copy` 操作将工程文件夹内的 `.json` 文件复制到 `transmission` 软件的 `settings.json` 文件中，完成对 `transmission` 软件配置信息的更新与更新。接下来打开 `transmission` 软件，并执行 `timer.py` 脚本执行自动做种程序。

3.6.5、其他

选写另外一些有必要叙述的部分，如模块的表现形式（以何种形式运行，如服务程序、应用程序、数据库脚本等等）、有无人机交互界面（有则给出其类型和设计原则）等。

3.7、种子评分模块

3.7.1、功能描述

对 `hit.py` 脚本爬取的种子库中的种子文件进行评分，评分的方法主要为默认评分、神经网络评分、传统回归方法评分。

3.7.2、接口描述

种子评分模块的代码名称为 `evaluate.py`。

输入记录了由 `hit.py` 脚本爬取的种子列表信息，输出一个 `list` 记录了种子的评分信息，具体的评分策略受 `.conf` 文件中的变量控制。

3.7.3、数据结构描述

本模块采用顺序存储，无特殊数据结构。

3.7.4、实现思路

给出实现该模块的基本思路，包括对模块内部结构、算法、编程方法等的初步设想。

默认评分根据的是种子的爬取顺序，因为 `hit.py` 爬取种子的顺序根据的是北邮人 `bt` 热门种子的热门程度进行爬取和下载的，该热门程度也是具有一定的评判标准，因此将其作为

默认评分使用。神经网络评分主要是根据由 hit.py 爬取到的种子的相关信息如下载文件大小、总下载量、当前做种人数、当前下载人数等信息通过训练神经网络来拟合一个可以用于评价种子好坏的函数，从而为各个种子计算对应评分。传统回归方法与上述神经网络方法类似，区别在于选用传统的机器学习方法如 SVC 进行非线性函数的拟合，从而为各个种子计算对应评分。

3.7.5、其他

选写另外一些有必要叙述的部分，如模块的表现形式（以何种形式运行，如服务程序、应用程序、数据库脚本等等）、有无人机交互界面（有则给出其类型和设计原则）等。

3.8 界面展示

3.8.1、功能描述

该模块完成展示与人机交互的功能。

3.8.2、接口描述

从.conf 文件中读取各种控制信息，将重要信息在界面上展示出来并用于界面初始化。

从 torrent_information 的 csv 文件中读取种子信息，将种子的关键信息在界面上展示出来。

将用户输入的用户名和密码传递给模拟登录模块实现用户登录功能。

3.8.3、数据结构描述

本模块采用顺序存储，无特殊数据结构。

3.8.4、实现思路

本模块的主函数完成以下三个功能：①配置信息的展示；②用户登陆与信息显示；③种子信息与下载进度的展示；④下载种子的开始、暂停和继续。

功能①中的各类信息包括：最大同时做种的数目、允许的最大下载空间、上传与下载速度限制、种子监视文件夹、资源下载文件夹。这些信息都从.conf 配置文件中获得。

功能②输入用户名与密码后将登陆请求发送给模拟登录模块进行登陆。

功能③中种子的各类信息包括：排名、资源名称、种子名称、下载类型、资源大小、做种数。这些信息都从 torrent_information 的 csv 文件中获得。下载进度通过一个函数进行实时监控并显示。

功能④对开始、暂停和继续三个按钮关联槽，实现种子的下载控制。

3.8.5、其他

选写另外一些有必要叙述的部分，如模块的表现形式（以何种形式运行，如服务程序、应用程序、数据库脚本等等）、有无人机交互界面（有则给出其类型和设计原则）等。

4、数据库与数据结构设计

4.1、数据库及数据表

本系统不使用数据库，以插件方式独立运行于个人电脑。部分数据存储以 txt 格式写入软件安装目录，此外还有部分软件运行必备的数据，例如验证码部分的校验字符库，将以编程语言易调用的格式写入安装目录的“.dat”后缀的文件。

4.2、数据结构设计

本系统内使用的数据结构主要再 txt 文件中的配置项，包括设置做种个数或区间范围、设置做种收益阈值、设置最大占用磁盘空间大小或区间范围以及系统运行时间或做种周期等

内容。

配置项按照层次关系使用标准的 json 格式进行存储，可根据编程语言转换为对象（object），纪录（record），结构（struct），字典（dictionary），哈希表（hash table），有键列表（keyed list），或者关联数组（associative array）等数据结构。

4.3、数据存储设计

为了使得软件作为小工具更加方便使用，系统数据不基于数据库存储，以 txt 和 dat 格式存储在软件安装目录中，读取方法按照不同编程语言遵从标准的 IO 流要求，按照模块区别构建不同存储文件。

在安全保密方面，软件不保存任何用户信息，包括但不限于用户名、登陆密码和模拟登陆的 cookies，对于 bt 种子文件中的 passkey 不做处理，我们认为用户个人电脑是相对安全的环境。

5、接口设计

4.1、外部接口

仅有一个，主要是软件操控开源的 bt 客户端软件，调用其功能的接口：

接口名称	Transmission 软件控制接口
接口地址	/etc/transmission-daemon/setting.json
输入方式	json
输入数据	<dict>
返回方式	0/1
返回数据	设置成功： 1
	设置失败 0

4.2、内部接口

验证码识别模块与模拟登录模块之间接口说明：

验证码识别模块封装在单独文件中，使用 python 编写，由 python 模拟登陆模块进行调用，采用函数调用方式执行接口，接口需要传入参数一个，为验证码图片在本地的存储路径（可以为绝对地址或者相对地址），模块的返回字段有三个，即三个输出参数，一个为 int 类型数值表征错误码，0 表示调用成功，非 0 表示调用失败，第二个字段为字符串表征调用信息，如果错误码非 0，则此字段返回错误信息，第三个字段为 data 部分，同样为字符串，当错误码为 0 时，有此字段，包含六个字符，为验证码识别结果。

其他接口的简单说明：

接口名称	种子评分接口
接口地址	./evaluate.py
输入方式	<dict>
输入数据	种子各项数据

返回方式	0/1
返回数据	传递成功: 1, Score
	传递失败 0

接口名称	更新做种列表接口
接口地址	./timer.py
输入方式	Json
输入数据	<dict>
返回方式	0/1
返回数据	传递成功: 1
	传递失败 0

接口名称	Bt 开源客户端控制接口
接口地址	./initial.py, ./gl.py
输入方式	Json
输入数据	<dict>
返回方式	0/1
返回数据	传递成功: 1
	传递失败 0

接口名称	模拟登陆模块—>验证码识别模块接口
接口地址	./login.py
输入方式	String
输入数据	验证码图片的相对存储路径
返回方式	0/string
返回数据	传递成功: 返回相对路径
	传递失败 0

接口名称	模拟登陆模块接口
接口地址	./login.py
输入方式	json
输入数据	与'https://bt.byr.cn/takelogin.php'的会话 session 的 cookies
返回方式	0/session.cookies
返回数据	传递成功: 返回会话 cookies 值

	传递失败 0
--	-----------

接口名称	爬虫模块接口
接口地址	./hit.py
输入方式	json
输入数据	执行信息获取的各 hit 函数的字符串返回值
返回方式	0/string
返回数据	传递成功： 返回各函数字符串返回值
	传递失败 0

接口名称	用户界面配置信息接口
接口地址	./main.cpp
输入方式	String
输入数据	.conf 文件中的配置信息
返回方式	0/String
返回数据	传递成功： 文件中的配置信息
	传递失败 0

接口名称	用户界面种子信息接口
接口地址	./main.cpp
输入方式	String
输入数据	.torrent_information 文件中的种子信息
返回方式	0/String
返回数据	传递成功： 文件中的种子信息
	传递失败 0

接口名称	用户界面登陆接口
接口地址	./main.cpp
输入方式	String
输入数据	用户输入的用户名与密码
返回方式	0/1
返回数据	传递成功： 1
	传递失败 0

6、其他设计

软件按照流程顺序执行，通常情况下仅考虑一个线程操作，后期考虑多线程进行信息爬取、种子下载以及评分算法计算。

在安全设计方面，软件封装后考虑运行前进行 hash 校验防止对代码的恶意篡改，软件同时不保存用户名密码等重要信息，避免带来安全问题。软件运行以插件方式进行，严格控制内存和 CPU 占用，避免影响系统正常使用。