为了获取人工智能及其相关领域的知识，我们选择了CSDN、博客园、简书、知乎等比较著名的社区。在选择编写爬虫的语言方面，我们选择了python。python不单用于抓取网页文档的接口简洁，同时其访问网页文档的API也相当完整(如urllib2库)；除此之外，很多网站会封杀爬虫。这时我们需要构造合适的请求来模拟浏览器的行为，而python提供了非常优秀的第三方包如Requests，mechanize。在抓取了网页之后，还需进一步的处理（比如过滤html标签，提取文本等）。而python的beautifulsoap库等能让我们编写非常简洁的代码即可完成大部分文档的处理。虽然一些其它语言或工具来实现以上功能，但是python是最快速简洁的语言。

在编写爬虫的时候也遇到各种各样的问题，记录如下:

1.robots.txt

robots.txt通常存放于网站根目录下，负责告知我们哪些内容是不能被获取的，哪些是能被获取的。其实际上并不是一个必须遵守的规范。如果爬虫不遵守，那么网站的隐私是不能通过这种方式来防止泄露的。

2.头文件headers

浏览器和爬虫程序，通常会向服务器发一个头文件：headers 以发送网络请求。这里面的大多数的字段都是浏览器对服务器的标识，因此大多数时候我们编写的爬虫，都需要伪造headers以掩盖爬虫的身份，向服务器发送请求。而符合该网站要求的headers，可打开浏览器开发者工具-^选择Network-^刷新页面-^选择最上方的文件-^找到request headers 即可

3.验证码

部分网站会要求输入验证码，应使用python的selenium库，模拟浏览器，网站会要求扫二维码进行验证，扫二维码即可解决，但验证码一直是反爬虫一个非常有效的手段，但其实这样作的工作量还是很大。

4.IP限制

封ip和封账号也是非常有效的反爬虫策略，此次项目在通过搜狗搜索引擎对微信文章进行爬取时，便出现了这种情况，封掉了IP，网页返回的状态码为302，解决方法:

(1)限制程序抓取频率，每隔几秒登录一次（这种方法工作量巨大）

(2)用多个账号、或者多台机器抓取，既处理了爬虫问题，也等同于做了分流处理，可以降低单台机器带宽压力（此方法在此次项目实现有点困难，不现实，仅供参考）

(3)使用代理IP，先写一个爬虫程序对免费代理IP的相关网站进行爬取。此爬虫程序较为复杂，这次选择的是使用在github上开源的程序(github上有详细的使用说明，以及需要的环境，非常方便，有效的解决了此次项目的问题)，虽然免费网站的代理IP，每一天能用的数量少，质量不高，爬取速度非常慢。但是也足以解决这次项目中IP被限制的问题

5.cookie

通过cookie限制抓取信息。 此问题在抓取博客园，csdn时均遇到，我们选择的是在伪造请求头时，因为headers里包含cookie，同时伪造cookie即可，通过浏览器的F12查看器，然后逐个尝试在整个过程都请求的URL(主要包括HTML、JS、XHR)，直到成功为止，一般情况下是最上方的HTML包含了我们所需要的cookie。

6.JS渲染

采用JS渲染页面。 有些时候返回的页面是由JS操作DOM得到，我们实际上拿不到正确数据。这个也是非常有效的反爬虫策略。我们选择使用selenium库模拟浏览器请求返回渲染完JS的页面。

综上，可以看出，即使编写爬虫时有非常多的困难，python都有多种库来帮助我们解决问题以得到初始数据。而接下来我们就将进行数据预处理