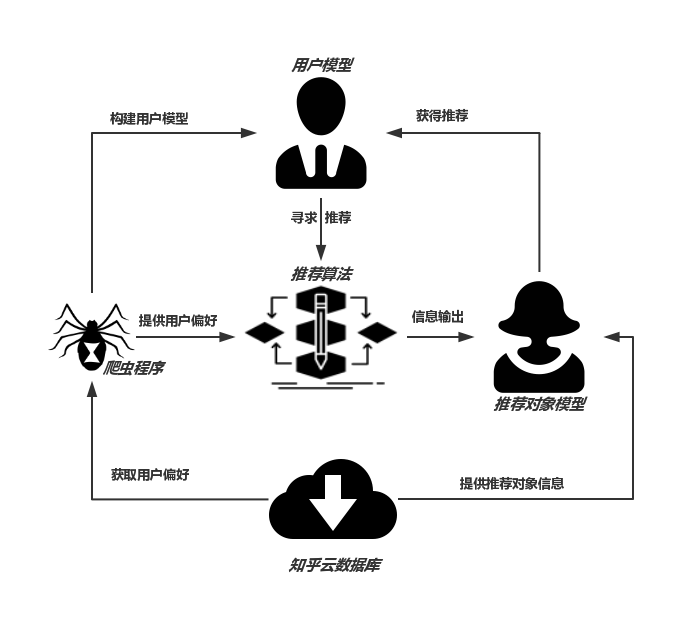
## 推荐系统

互联网的出现和普及给用户带来了大量的信息，满足了用户在信息时代对信息的需求，但随着网络的迅速发展而带来的网上信息量的大幅增长，使得用户在面对大量信息时无法从中获得对自己真正有用的那部分信息，对信息的使用效率反而降低了，这就是所谓的信息超载（information overload）问题。

无论是微博、论坛，还是贴吧、博客，信息超载的问题都非常常见，解决信息超载问题一个非常有潜力的办法是推荐系统，它是根据用户的信息需求、兴趣等，将用户感兴趣的信息、产品等推荐给用户的个性化信息推荐系统。

其实推荐系统在各种社交网站、购物网站都已经非常常见了，知乎中也存在着这样一个推荐系统，每天为你推送一些你最近感兴趣的问题、话题等，知乎所使用的推荐算法我们无从得知，但是我们可以自己根据平时使用知乎的习惯与经验来自行设计一个推荐系统。

系统框架如下：



系统运行后，输入用户ID即可通过爬虫程序获取知乎云数据库中存储的用户信息，程序通过用户信息构建用户模型，并将用户偏好输入到推荐算法中，推荐算法根据用户偏好计算出适合推荐的对象模型，然后从知乎云端数据库中获取推荐对象的信息，最后提供给用户。

构建用户模型时，具体到知乎社交网络，我们可以用以下信息集合来表示用户的偏好：

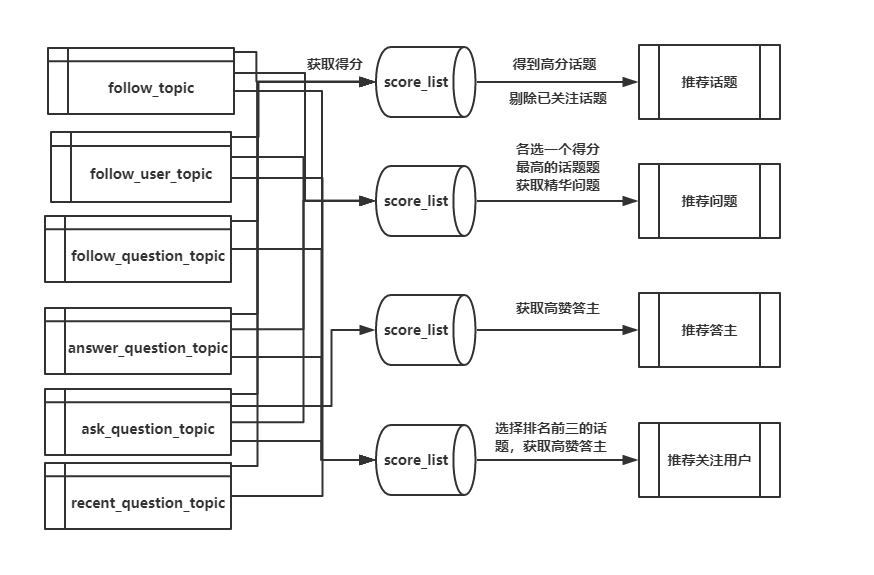
其中， 代表用户所关注的用户列表，代表用户所关注的问题列表，代表用户所关注的话题列表，代表用户回答的问题列表，代表用户提出的问题列表，代表用户最近浏览的问题列表。这些信息基本上包含了一名用户在知乎社交网络中的大部分偏好信息。

根据以上偏好信息，我们可以构建出一个用户模型，之后我们对用户的这些偏好信息进行加工处理：

* 对于用户所关注的用户，我们利用爬虫程序，获取他们所关注的问题、回答的问题、提出的问题，提取出这些问题所属的话题，加上他们关注的话题，得到一个话题列表[ follow\_user\_topic ];
* 对于用户所关注的问题、回答的问题、提出的问题、最近浏览的问题，我们提取这些问题所属的话题，得到四个话题列表：[ follow\_question\_topic]、[ answer\_question\_topic]、[ ask\_question\_topic]、[recent\_question\_topic]，另外，用户所关注的话题自成一个列表[ follow\_topic];
* 根据以上两个步骤，我们即可得到6个话题列表，然后我们给这6个不同的话题列表人为地设定6个权重值为{}，然后对遍历所有话题，统计每个话题的总得分，得到话题和得分列表；

得到上述话题—得分列表后，我们将根据各项属性，向用户推荐各种不同的内容，具体操作如下：

* 向用户推荐话题：从话题得分列表中剔除掉所有用户已关注的话题，然后选择剩下话题的前8名作为推荐话题推送给用户；
* 向用户推荐问题：从[ answer\_question\_topic]、[follow\_topic]、[follow\_user\_topic]、 [ask\_question\_topic]、[recent\_question\_topic]五个话题列表中，各选一个得分最高的话题，再获取这些话题下的精华问题，最后将这些问题推荐给用户；
* 向用户推荐答主：从用户所提出的问题所属的话题列表[ask\_question\_topic]中选择几个高得分的话题，再爬取这些话题的精华问题的高赞（优秀）答主，将他们推荐给用户；
* 向用户推荐关注用户：从所有话题得分中选择排名前三的话题，获取这些话题下的精华问题的优秀答主，推荐给用户；

推荐算法流程图如下：  


* 结果展示：  
  （待补充）
* 总结：

上述推荐系统主要是基于用户的兴趣、关注来提供推荐信息，在给话题进行评分时，权重值都是我们人为根据平时的使用习惯来设定的。原本我们拟定的方案是从知乎上爬取大量用户的上述信息，采集数量较多的样本，搭建一个神经网络来训练这些权重值，但是由于知乎有反爬虫措施，要爬取大量样本的难度太大，时间也需要很久，因此我们最后只能暂时搁置此方案，而人为地设定权重值，通过几个样本手动调整这些权重值。虽然如此，但是我们的推荐系统效果其实还不错，基本上都能推荐出用户感兴趣的内容。

我们的推荐系统是一个在线推荐系统，所有的数据都是通过爬虫现用现爬的，之前用于分析社交网络的那个数据集太老，无法用于我们的在线推荐系统。假如我们能够轻松获取知乎的大量数据，我相信通过机器学习的方式，一定能训练出一个很棒的推荐系统出来。