**reculike发布**

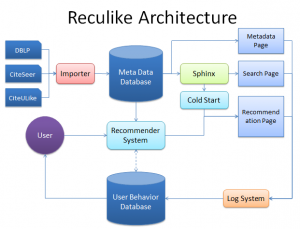
一直认定推荐引擎将会成为互联网下一个十年的主导，一如搜索引擎在上一个十年的表现一样。

我们笃信推荐引擎的巨大潜力，为了向movielens致敬，7月中，我们启动了paperlens，目前已有了原始版本，可以通过reculike.com访问。

非常荣幸我能参与这个从零开始的开源项目，整理了一些reculike相关的信息，希望对想了解reculike的同学们有所帮助。  
**0. Why build reculike?**  
xlvector正在写一本关于推荐系统的书，书中需要有真实的推荐系统及源码，所以我们创建了reculike.com。

**1. reculike是什么？**  
reculike是一个开源的item-based的论文推荐系统。  
on Reculike.com：  
1）We recommend paper to you，based on your behavior(from click to recommend、search or view behavior).  
2）You can search what you want, and show your feeling(like/dislike) about the results.  
3）You can view papers, find related papers and authors, and show your feeling about these.  
4）you can connect users and experts.

2.概要  
reculike架构图：

[](http://yechoul.com/michael/wp-content/uploads/2011/08/resys-arch1.png)

Reculike Architecture

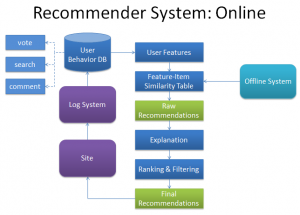
All Sub-Systems:  
Importer  
Search Engine  
Recommender System  
API/Site

2.1. Importor  
该模块主要是数据导入功能，全部用python编写。  
目前的数据来源为ACM、ArNet、citeseer和DBLP。

2.2. Search Engine  
该模块主要为整个系统提供搜索服务，based on sphinx2，API用PHP编写。  
目前主要实现了两块：search和suggest。  
对于search，我们的基本排序策略是：保证相关度的情况下，尽量返给用户citation较多，较新的paper。  
排序公式：@weight\*log2(3 + citations/1000) /(2030 – year)  
其中@weight代表相关度，citations是paper的引用次数，year是paper发表时间。

2.3. Recommender System  
为reculike的核心，推荐引擎模块，用PHP编写。  
推荐引擎主要分为两个部分：Off-line System和On-line System  
2.3.1. Off-line System:  
computing paper similarity table.  
即用于计算paper之间的相似度。  
这些similarity table可以有很多个，目前我们想到了三种：  
content-based similarity(title,author,summary,keywords…)：基于paper内容的相似度表。  
citation-based similarity：基于citation的相似度表，两篇paper引用的文章越相似，我们就初步认为这两篇paper的相似度越高。  
collaborative filtering：熟知的协同过滤。

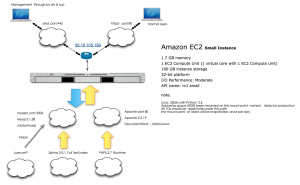
2.3.2. On-line System：  
Making recommendation by users’ historical behavior.  
即根据用户的历史行为数据作出推荐。  
推荐引擎的架构图：

[](http://yechoul.com/michael/wp-content/uploads/2011/08/resys-online.png)

reculike on-line recommender system

2.4. API/Site  
其中API下提供了recommendation、search、auth等API，供其他外部系统调用用以实验。我们将在以后提供API文档。  
Site即看到的reculike.com内容，PHP编写。  
主要有以下features：  
recommender system、passport、search engine、view papers and their relationships等。

**3.服务器**  
我们使用了Amazon EC2（http://aws.amazon.com/ec2/）的小型实例。  
默认配置了1.7 GB的存储器，一个虚拟核心上运行一个EC2 Compute Unit（一颗processor:Intel(R) Xeon(R) CPU E5507 @ 2.27GHz）,160GB的硬盘空间（貌似我们只有50G）, 32位平台（以上信息来自百度百科http://baike.baidu.com/view/6102577.htm）。

[](http://yechoul.com/michael/wp-content/uploads/2011/08/ec2.png)

ec2-small instance

**4.下一步我们将要做什么**

0）数据的完善：增加下载链接等  
1）site增加QA（仅支持英文）、social功能、意见反馈  
2）推荐增加reason

**5.关于团队**  
非常非常崇拜xlvector(http://xlvector.net/blog/)，reculike中绝大部分内容都是由xlvector亲自设计和编写的；另外还有一位默默coding的wangxing同学，第一次聚会我迟到了，所以还不认识他:(， 只知道我们是校友，没有任何言语上的交流，但却能感觉到他的努力。他们两个的勤奋一直让我自惭形秽。