2020-03-11

时间: 15:00～16:00

与会人员: 陈耿阳 田晨江 赵文祺 孙逸伦

记录人: 陈耿阳

会前准备

迭代一中的一些点

- 文档组织
  - 启动文档、计划文档
  - 需求规格说明
    - 适当裁剪，但在迭代三需要拿出完整的，带性能指标的规格说明，因为届时是面向最终用户的，需要一个更加专业的产物
    - 原型是必要的，最好能有交互性，但是没有也无伤大雅
  - 设计文档
    - 接口文档是必要的
    - 可以预料到，迭代二三中架构会有大幅改动，这个时候维护架构文档就有必要了
      - 分布式系统的一些设计细节有必要进行记录
      - 迭代一纯粹是因为要求凑字数的，单体的B/S架构有啥好写的
    - 详细设计文档没有意义
      - 主要是作业要求，小规模团队纯属浪费时间
      - 可以以接口文档为蓝本，适当补充一些细节
  - 测试文档
    - 主要是倒逼自己思考功能上有没有缺陷和遗漏，**发现错误**
      - 不是为了追求指标，刷覆盖率或者通过率，也不是为了显得自己的代码质量很高
      - 什么样的用户都是存在的，尽可能往鬼畜的地方想
      - 以发现错误为第一标准去“挑刺”
    - 前端单元测试适当编写，后期手工测试为主
      - 人力有限，强交互的场景耗时很多，影响核心业务的开发
    - 后端单元测试一定要写
  - 小组总结
    - 有话则长，无话则短
    - 真诚一点
- 因此，**迭代二**应该拿出的产物：

    - 启动文档
    - 计划文档
    - 需求规格（用例、**原型**）
    - 设计文档（架构设计、详细设计、**接口文档**）
    - 测试文档
    - 小组总结
- 工作流程上
    - 迭代一的git flow过于僵硬，带来了一些不好的体验
        - 在本次迭代中进行改进，参见启动文档
        - 在此做自我检讨
    - 整体的大方向应该是对的，文档边写边改，以用例为单位进行划分
        - 有什么可以改进的地方吗？现有的模式是不是有点僵硬？都可以改。

用户希望看到的：DRUSS

- 有用：我们要给用户传达一个什么价值主张？
    - 我个人倾向于就做一个推荐系统
        - 迭代一：用户数据检索 + 管理员数据导入
        - 迭代二：用户实体建模（实体画像、实体关系） + 管理员数据管理
        - 迭代三：用户热点推荐（基于用户兴趣和历史记录）+ 管理员自定义数据源和脚本，知识图谱的建立
    - 怎么才能真正有用？

        a) 你有考虑过每个研究者在不同论文中的姓名字符串不同吗？

        b) 你有考虑过相同组织中的同名学者如何区分吗？

        c) 你有考虑过一篇论文如果 5 个作者，应该怎么区分他们的贡献吗？

        d) 你有考虑过对每一篇论文的参考文献再做挖掘吗？

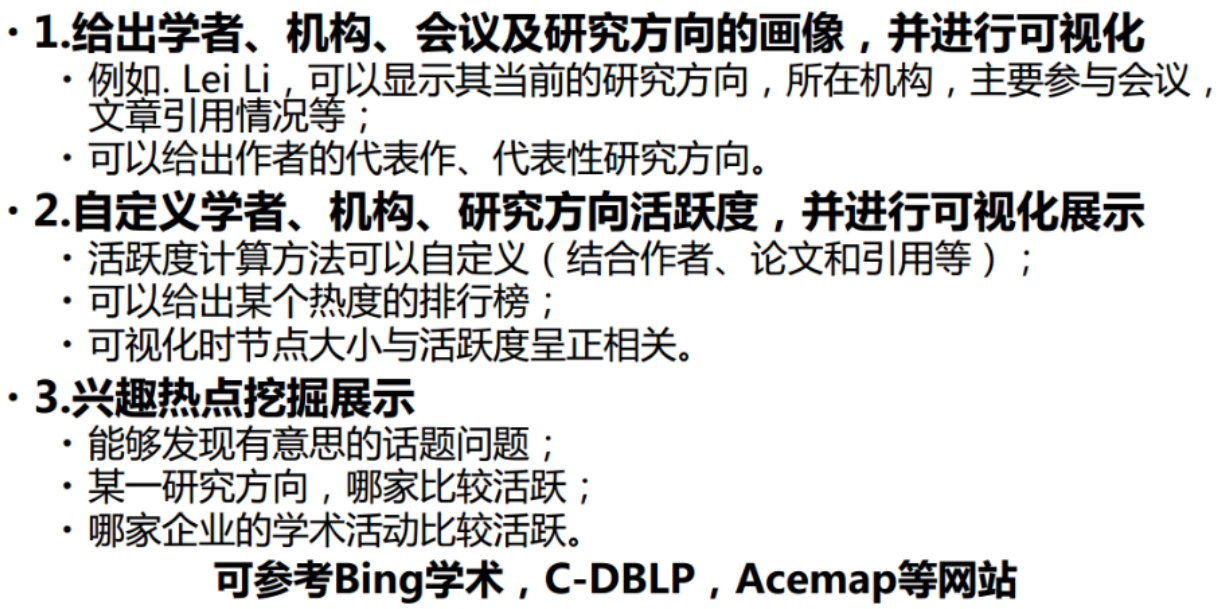
        e) 你有考虑过在一个人和人的网络之中，怎么辨别他们不同的价值吗？

        f) 你有考虑过选择导师，你要选择一个网络中的中心节点还是连接节点吗？

        g) 怎么能分析出软件工程领域研究分类和热点变化？
- 可靠：少出bug
    - 功能简单出bug的概率也比较低，某种程度上是好事
    - 测试倒逼可靠性，再次强调测试的目的是**找茬**，而不是证明自己的系统没问题
- 好用：用户体验
    - 高性能，这个是显然的
        - 架构需要调整：负载均衡？或者分布式系统？
            - 数据量上来了怎么办？还能保持现状吗？
        - 懒加载？SSR？总之接口需要调整。
    - 高可用
        - 6+希望看到**高并发**场景下仍然可用

        - 如果服务器宕机了怎么办?
    - 交互体验好
        - 搜索关键词应该高亮
        - 进一步优化，参考迭代一心愿单
- 数据安全：数据一致性
    - 高性能情况下分布式是必然的，那么如何保证数据一致性?
- 系统安全：赵桑の奇妙矿机
    - 虽然暂时不考虑，但是如果真的出现了，应该如何应对?

迭代二检查点：



Zwq

今天我看了一天的数据，心情十分跌宕起伏。早上我看爬虫崩了，然后起来用橙酱的服务器又部署了一个爬虫，看着还爬的蛮快，心里美滋滋。

下午的时候开始看知识图谱，图学习和推荐系统，一不小心，看到了人家居然有现成的数据，还TM有12个G，辣针滴nb，觉得做图嵌入大有希望。然后我满心激动地等了两个小时下载，又花了一个多小时想办法弄进数据库。**（在导入本地mongo的时候失了智，一不小心把远端的爬虫数据库给删了，啊啊啊啊啊啊，我4w条数据啊）**

晚上仔细检查数据库的时候发现坏了，这个12个G的数据和我想的似乎有很大差别。

这个数据主要是为了研究学者、文章、会议之间的研究关系的，虽然关系很全面也有id标注，但重大问题是这300W条数据TMD的**abstract和url全是NULL**，也就是这是纯图关系，不带内容信息的那种。然后我盯着缓慢蠕动的爬虫又一次陷入了自闭。

1. 我认为我们做出一个成熟的知识图谱推荐系统是相当困难的

根据我今天下午看到的文章，大体把现有的推荐系统分成两种

- 大公司的推荐系统。他们往往都是有足够的用户基数，使得自己的系统可以通过采样用户的轨迹来进行图表示的学习。

- 研究人员的推荐系统。大多数论文使用的数据集，都满足两个条件中的一个，**同构的**，或者**半监督**的，一般是做实体分类（比如把论文分到合适的会议中）。

很不幸，我们手上的数据集

- 首先是异构的（文章、作者、会议），包含多种实体，这意味着我们不能通过同构图中的Deepwalk等经典非监督方法来学习图表示。
- 其次，我们没有足够的用户基数来启动我们的学习。到时候助教们看到的会是个冷启动的推荐系统。

2. 我认为我们的关注点不应该，或者是不应全放在数据正确性上

从爬虫到数据清洗到纠错、众包，每一步都是大量的人力成本。我想，我们都不愿意天天对着一堆JSON👀干瞪眼，从里面找出问题并标注。

清洗数据当然是很重要的，我们应该对**数据进行一个大体上**的处理，而那些更加琐碎的细节，其实这些都是可以用更加廉价的劳动力去完成的。**我觉得，我们应该把宝贵的时间花在真正有意义的事情上，去优化研究人员的论文查阅流程，去优化管理人员管理系统的体验。授人以渔才能体现作为软件工程师的价值**。

3. 我们可以做面向科研领域的狭义的“推荐”

虽然论文看的不多，但也有一定心得，虽然我们做不了广义上的推荐，但我们可以做面向科研领域的“知识推荐”。抖音成功的地方就在于，他可以给你一直推送新鲜的动态，本质上连用户搜索都不需要，给用户产生了极大的粘性。但科研领域我想应该不会有人去一直漫无目的地刷论文动态吧，这也引出了假设：**用户在使用论文系统的时候是抱有一定目的的，至少，不应当是毫无目的的。**

通常，科研人员所关注的，是自己所在的一个较为细分的领域，同时也会对相关领域有一些关注。拿我今天下午打比方，想去查关于图学习方向的论文，这是我之前从未了解过的方向。那么作为一个萌新，我该快速理清头绪，看一些关键性的论文呢？一个比较朴素的想法就是，上知乎看看有没有大佬总结过这方面前人做过的较为有代表性的工作，然后跟随这个**论文清单和论文本身的引用**，一篇篇慢慢看。

1. 比如，图表示学习中的开篇之作是Deepwalk的那篇文章，现在有2k引用了。
2. 再后来，有了node2vec，对Deepwalk的搜索方法进行了改进
3. 2017年有metapath2vec，在另一个维度上开辟搜索路径

4. 2017年也有研究者借用CNN中参数局部性的思想，搞了个GCN，同样大获成功
5. 。。。。

也就是说，研究领域本身就是有很强顺序性的，后人往往是在前人的工作基础上进行优化和改进。

> 我的想法是，可不可以把这个研究领域前后发展的时间顺序树🎄，以一种直观的方法展现给科研工作者，使他们可以快速了解本领域内那些论文是重要的，从而帮助他们快速上手

可以使用的数据

- 文章关键字，指明研究领域。
- 引用顺序树，**为什么说是树呢，因为引用关系中是不可能包含环的，因此是一个有向无环图。树结构也能够很清晰地反映出一个领域蓬勃发展的情况。**
- 文章引用量，这个领域就是从这篇文章开始火起来的。

4. 我们可以参考文献管理软件papers，做方便的论文导入

通常，让管理人员手动写json或者csv来添加文章是很让人难受的，我们可以用OCR来改变这一困难，实现高效论文导入。（当然我只是看到别人有，不知道实现起来是否困难）

- 管理员直接导入pdf / 文章链接，后台通过OCR识别title，author等信息直接导入
- 使用chrome插件，当管理员在chrome中查看一个pdf文档时，提示是否导入数据库

我觉得，多从用户的角度考虑问题，提出**有意义、可实现**的需求，而不是像其他组上来就大数据、机器学习、人工智能、分布式疯狂贴金更有意义。

讨论内容

1. 前端基于迭代一优化：ts=>js+SSR
2. 迭代二：
    1. 2->4个排名
    2. 三个主页
        1. 作者主页
        2. 机构主页
        3. 研究方向：Roadmap
3. 下次会议时间：周六上午 9:00