# 第三章 基于时间树的搜索行为数据收集方法

## 3.1复杂搜索过程管理问题

搜索引擎已经成为用户解决日常生活中遇到问题的最重要的信息来源之一。当搜索任务简单时，现有的搜索引擎已经能够很好理解用户的信息需求，并返回给用户包含能够满足用户信息需求的搜索结果。例如，当用户想知道北京的当天的天气情况时，只需要在搜索引擎中输入“北京 天气”，搜索引擎便能直接返回北京当天的天气情况。此时，用户通过查看相关网页便能很快获得关于北京天气的信息。甚至，当用户需要了解当地的天气情况时，只需要在搜索引擎中搜索“天气”，搜索引擎便可以了解到用户是可能是想了解当地的天气情况，并直接返回与本地天气相关的资源，从而满足用户的信息需求。

然而，当搜索任务更加复杂时，用户往往需要在搜索引擎中进行多次、往复、大量、长时间的搜索和点击，并逐渐逼近用户的搜索目标。一种典型的情况是，当用户希望撰写一个领域或问题的综述文章时，用户往往对该领域或问题不是特别熟悉，从而需要在搜索引擎中进行大量查询和点击，以逐步了解每个解决此领域或问题的方法的具体内容和优劣情况，最终形成此综述文章的主体内容。例如，用户想要写一篇关于解决0-1规划问题的方法综述，用户需要了解什么是0-1规划问题，解决0-1规划问题有哪些具体的方法，这些方法是怎么执行的过程，以及对比这些方法以及这些方法求得的解的质量和所花费时间。但是用户对于0-1规划问题和解决0-1规划问题的方法不太了解。因此，根据用户的对于0-1规划问题的现有经验和搜索经验，形成了如下的搜索过程：

* 1. 用户首先需要了解什么是0-1规划问题。所以，用户搜索了“0-1规划”，通过点击搜索引擎返回的与0-1规划相关的一些网页从而建立了对0-1规划的概念。
  2. 用户想要了解 0-1规划现有哪几种解决方法，于是用户在搜索引擎中查询“0-1规划的解决方法 分类”，尝试点击了几篇文章或网页之后，用户只是找到部分解决方案，而并不能获取到所有的解决方案。
  3. 基于步骤2，用户希望扩大一下搜索范围从而在搜索引擎中查询了“0-1规划 解决方法”，并继续点击了几篇文章或网页之后才了解到0-1规划大概包括以下几种解决方法：隐枚举法、分支定界法、割平面法、罚函数法、拉格朗日松弛法以及现代启发式算法。
  4. 用户想要继续了解哪些现代启发式算法能够解决0-1规划问题，因此用户在搜索引擎中查询了“0-1规划 启发式算法”，并点了一些搜索引擎返回的对应的文章或网页，了解到可以用遗传算法、差分进化算法、粒子群算法以及蚁群算法等启发式算法去解决0-1规划问题。
  5. 用户想要了解遗传算法是怎么解决0-1规划问题的以及遗传算法在解决0-1规划问题的质量问题。因此用户查询了“遗传算法 0-1规划”，了解到遗传算法又包括基本遗传算法和许多改进后的遗传算法。
  6. 用户又分别对这几种遗传算法进行了搜索以了解它们之间的区别联系。
  7. 与上述步骤类似，用户将进行大量搜索来获取其他0-1规划算法的信息。

从上面用户对于0-1规划的例子中可以看出，用户完成复杂搜索需要进行大量的查询和点击。在进行复杂过程中，用户需要对点击、查询和子任务进行记录和管理。然而现有的搜索引擎并不能对搜索过程记录和管理提供支持。具体表现在：

【问题1】现有搜索引擎不支持对点击的记录和管理。

在复杂搜索中，用户需要进行查阅大量资料，因此用户会产生很多点击来查看相关内容。用户所需要查看的内容并不是全部有用的，用户需要对包含有用内容的点击进行记录和管理。根据点击内容作用的不同，此问题又包含两个子问题：

【问题1.1】现有搜索引擎不支持对包含支持用户进行下一步搜索的信息的点击记录和管理。例如上述搜索过程中的步骤（4）中查询关键词“现代启发式算法”搜索便来源于步骤（3）中点击中的信息。对此类点击以及点击中的信息进行记录和管理都是必要的，因此，此问题包含两个子问题：【问题1.1.1】现有搜索引擎不支持对此类点击记录和管理；【问题1.1.2】现有搜索引擎不支持对此类点击中包含支持用户进行下一步搜索的信息记录和管理。

【问题1.2】现有搜索引擎不支持对包含支持用户完成搜索任务的信息的点击记录和管理。如上述搜索过程中的步骤（5），用户通过点击获取到遗传算法解决0-1规划问题的方式的信息，而用户并不需要根据此信息进行后续搜索。此问题同样包含两个子问题：【问题1.2.1】现有搜索引擎不支持对此类点击记录和管理；【问题1.2.2】现有搜索引擎不支持对此类点击的作用记录和管理。

【问题2】现有搜索引擎不支持对查询的记录和管理。

用户在复杂搜索过程中存在大量查询，而每种查询对于用户带来的收益是不同的。对有收益的查询进行记录和管理帮助用户回顾之前获得的收益及完成的任务，基于此组织后续搜索。对无收益或收益很少的查询进行记录可以帮助用户避免同样的搜索。因此，此问题又可拆分成两个子问题：**【问题2.1】现有搜索引擎不支持对查询的记录和管理**；**【问题2.2】现有搜索引擎不支持对查询带来的收益记录和管理**。

【问题3】现有搜索引擎不支持对子任务的记录和管理。

在复杂搜索中，搜索者往往把搜索任务划分为多个子任务，然后对子任务进行分别搜索。如上述例子中，搜索者把解决0-1规划的方法划分为隐枚举法等。将任务划分为子任务的优势是显然的，用户可以逐步解决子任务，减少搜索任务的遗漏，搜索更加有连续性和组织性。对子任务进行记录和管理帮助用户明确当前子任务的完成进度，基于此对子任务进行补充查询从而最终完成搜索任务。然而在复杂搜索过程中，用户几乎不可能完全完成一个子任务再去搜索另一个子任务。因此，现有搜索引擎不支持对子任务的记录和管理。

现有搜索引擎在支持复杂搜索存在上述不足，导致用户在进行复杂搜索时需要大量的记录和管理，并由此给用户带来的额外的负担，这些负担需要一种全新的搜索过程管理方法进行解决。

## 3.2基于时间树复杂搜索过程管理方法

为解决上述在复杂搜索中存在的问题，本文提出了一种可视化的复杂搜索管理方法-基于时间树的复杂搜索管理。本节将给出时间树的定义以及时间树解决上述复杂搜索中存在的问题的方式。

### 3.2.1 时间树形式化描述

图3.1 时间树示例

Fig. 3.1 TimeTree Example

如图3.1所示，时间树将用户提交的查询词和点击的搜索结果按照时间顺序以树的形式组织起来，形成了对整个探索式搜索过程的可视化描述，方便用户随时回顾整个搜索过程，进而规划下一步的搜索。时间树是由查询节点和点击节点两种基础元素进行构成，其中，圆形为查询节点，方形为点击节点。下面给出时间树及其相关内容形式化描述：

**【定义3.1】查询节点（Query Node，****）：**搜索者进行查询后形成的节点。定义为三元组，。其中，为节点标志(Node Symbol)，为查询节点标志(Query Node Symbol)，在中，恒等于。在时间树中，使用图形○代表。为查询关键词（Query Keyword）。定义3.2中给出的形式化描述。

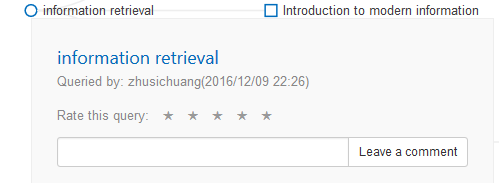
**【定义3.2】查询节点提示信息（Query Node Tooltip，）：**记录了查询节点相关信息提示框。被定义为一个五元组，,其中为提示信息可见性（ToolTip Visibility），在默认情况下 ，即为不可见，当搜索者使用鼠标在指向持续一秒后，才为可见，此时。为用户查询的时间(Query Time)。下面分别给出、以及的形式化描述。图3.2是查询节点提示信息示例。

图3.2 查询节点提示信息示例

Fig. 3.2 Query Node Tooltip Example

**【定义3.2.1】查询链接（Query Link，）：**重定位到原查询网页的链接。被定义为一个二元组，。其中，为查询链接文本（Query Link Text）, 为查询链接地址（Query Link Adress）。在时间树中，,通过点击可以直接再次查询。

**【定义3.2.3】查询评分（Query Rate，）：**查询的评分。查询评分的取值为空或者为区间[1,5]任意整数,因此定义为如下形式：。

**【定义3.2.3】查询评论集合（Query Comment Collection，）：**查询评论的集合。，其中为查询评论（Query Comment），为查询次数的数量。用户需要在中主动提交评论来增加查询评论。

**【定义3.3】查询节点集合（Query Node Collection，）：**查询节点的集合**。** ，其中为查询节点的数量。

**【定义3.4】点击节点（Click Node，）：**对搜索页结果进行点击后形成的节点。为三元组，。其中，为节点标志(Node Symbol)，为点击节点标志(Click Node Symbol)，在中，恒等于。在时间树中，使用图形□代表。为点击内容标题（Click Content Title），定义3.5中给出形式化描述。。

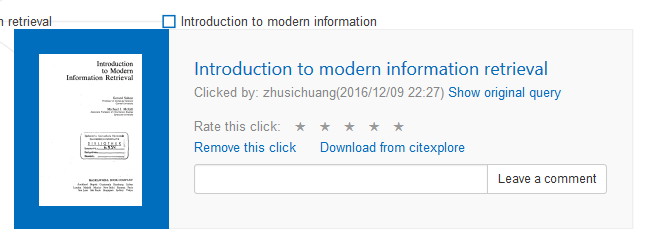
**【定义3.5】点击节点提示信息（Click Node Tooltip，****）：**记录了点击节点相关信息提示框。被定义为一个六元组，,其中，与中的为同样的含义与作用。 为用户点击的时间（Click Time）, 为点击文章缩略图（Click Article Thumbnail）。定义3.5.1、定义3.5.2、定义3.5.3分别给出、、形式化定义。图3.3是点击节点提示信息示例。

图3.3 查询节点提示信息示例

Fig. 3.3 Click Node Tooltip Example

**【定义3.5.1】点击链接(Click Link，)：**重定位到点击页面的链接。被定义为一个二元组，。其中，为点击链接文本(Click Link Text), 为查询链接地址（Click Link Adress）。在时间树中，,通过点击可以直接跳转到点击内容的下载地址。

**【定义3.5.2】点击评分（Click Rate，）：**点击的评分。点击评分的取值为空或者为区间[1,5]任意整数,因此定义为如下形式：。

**【定义3.5.3】点击评论集合（Click Comment Collection，）：**点击评论的集合。点击评论集合定义为如下形式: ，其中为查询评论，为点击评论的数量。用户需要在中主动提交评论来增加查询评论。

**【定义3.6】点击节点集合（Click Node Collection，）：**点击节点的集合。点击节点集合定义为如下形式: ，为点击节点的数量。

**【定义3.7】时间树节点关系（Relation，）：**时间树上节点之间的关系。被描述为一个三元组,其中（parent Node）为时间树上（Child Node）的父亲节点 。为与的关系，。下面给出，，的定义描述。

（1）**点击来源于查询（Click From Query，）：**当且时，。此描述意味着当用户的点击为查询的子节点时，点击是来源此查询的结果页。

（2）**查询来源于点击（Query From Click，）：**当且时，。此描述意味着当用户的查询为点击的子节点时，查询来源此点击中的内容。

（3）**查询之间的关系（Relation between Query and Query，****）：**当且时，。本文将查询之间的关系分为以下4种：查询细化、查询扩展、查询重构以及主题转移。关于查询之间的关系的讨论将在本文第五章展开。

**【定义3.8】时间树节点关系集合（Relation Collection，）：**节点关系的集合**。**,其中根据树的性质，时间树节点关系集合大小比节点数量小1。

**【定义3.9】时间树（TimeTree，****）:**将查询节点和点击节点按照时间顺序组织的树结构**。**时间树描述为一个三元组，。

时间树由时间树查询节点集合、点击节点、以及节点之间的关系集合构成。综上，时间树及其组成部分的所有定义已经全部给出。

### 3.2.2 基于时间树的搜索过程管理

本节描述基于时间树的搜索过程管理，并在此基础上给出时间树解决现有搜索引擎在进行复杂搜索存在的问题的方式：

（1） 用户使用点击节点来记录点击的相关内容。当用户在搜索引擎中进行点击，会在时间树中自动生成对应的点击节点，其中为点击中内容标题，基于此解决了问题1.1.1和问题1.2.1中对点击的记录和管理。用户在点击节点提示信息中记录点击中支持用户下一步搜索的信息或者支持任务完成的信息，并根据该文章的重要程度对点击进行打分，点击标志会随着分数的不同呈现出不同的颜色，用户可以很轻易在时间树上发现对自己有用的节点。基于此解决了对问题1.1.2和问题1.2.2中对信息的记录和管理。

（2） 用户使用查询节点来记录查询的相关信息。当用户在搜索引擎中提交查询词时，会在时间树中自动生成对应的查询节点，其中等于提交的查询词，基于此解决了问题2.1中对查询的记录和管理。用户在查询节点提示信息中同样可以记录点击带来的收益并进行评分。时间树提供了一种更好的方法来记录和管理查询带来的收益，查询节点的子点击节点中所记录的收益可以认为是查询带来的收益。基于此解决了问题2.2中对查询带来的收益进行记录的管理。

（3）时间树由查询节点和点击节点以及节点之间的关系构成。根据，用户的子点击节点与父查询节点服务于同一个子任务，用户的子查询节点与父节点往往也服务于相同或近似的子任务，因此服务于同一个子任务的查询节点和点击节点在时间树以簇的形式呈现。例如，如图，蓝色底纹的查询节点和点击节点服务于子任务“中国PM2.5”，橙色底纹的查询节点和点击节点服务于子任务“PM2.5危害”。基于此解决问题3中对子任务的记录和管理。

### 3.2.3 时间树生成算法

本文在3.1.1节给出了时间树的形式化定义以及在3.2.2中给出了基于时间树的搜索过程管理。为减轻用户使用时间树进行搜索过程管理的负担及增加搜索过程管理的准确性，本节给出时间树生成算法自动构建时间树。

在定义3.9中，时间树描述为由查询节点集合、点击节点集合以及节点关系集合组成的三元组。首先，本文给出时间树节点（查询节点和点击节点）生成算法，详细描述了时间树节点的生成以及节点的初始化：当用户提交查询词时，生成查询节点，然后初始化*NS*、*QK*以及*QNT*；当用户进行点击时，生成点击节点，然后初始化*NS*、*CCT*、*CNT*。具体的初始化值在下面算法中给出，此处便不再赘述。

表3.1 时间树节点生成算法

Table. 3.1 The algorithm of creating timetree node

|  |
| --- |
| **算法3.1**时间树节点生成算法 |
| 算法输入：用户操作action  算法输出：时间树节点*N*  1. **Begin** newTimeTreeNode(action):  2. **If** action.type == “query" then  3.  *QN* qn = new *QN*  4. qn.*NS* = QNS  5. qn.*QK* = action.queryKeyWords  6. *QNT* qnt = qn.*QNT*  7. qnt.*TV* = *FALSE*  8. qnt.*QT* = action.getTime  9. qnt *.QL.QLT* = action.queryKeyWords  10. qnt.*QL.QLA* = action.linkAddress  11. qnt.*QR* = NULL  12. qnt.*QNC* = ∅  13. *N* = qn  14. **Else** if action.type = “click” then  15. *CN* cn = new *CN*  16. cn.*NS* = *CNS*  17. cn.*CCT* = action.clickContent.getTitle  18. *CNT* cnt = cn.*CNT*  19. cnt.*TV* = *FALSE*  20. cnt.*CT* = action.getTime  21. cnt.*CAT* = action.clickContent.getFirstPageThumbnail  22. cnt.*CL.CLT* = action.clickContent.getTitle  23. cnt.*CL.CLA* = action.linkAddress  24. cnt.*CR* = NULL  25. cnt.*CNC* = ∅  26. *N* = cn  27. **Else then**  28. *N* = NULL  29. **End If**  30. return *N*  31. **End** |

定义3.7给出了时间树节点关系：时间树节点关系由父节点、子节点以及父子节点关系组成，其中父子关系被划分为点击来源查询、查询来源于点击以及查询之间的关系。为实现自动构建时间树，在生成点击节点时，对点击中的文章内容进行提取，并进行全文索引。在给出时间树节点生成算法后，下面给出时间树节点关系生成算法3.2。时间树节点生成算法把新生成的时间树节点与时间树原有节点建立关系。根据时间树子节点类型的不同，时间树关系生成规则如下：

1. 当子节点为点击节点时，算法首先获取点击来源的查询节点作为其父节点，并与之建立查询来自点击的关系。
2. 当子节点为查询节点时，算法首先将查询关键词进行分词处理并剔除其中的停止词，然后根据点击的内容索引求得关键词分词匹配个数mc。针对最高匹配个数mmc是否为0，本文采取两种不同的关系生成策略：(1)当mc为0时，算法选取时间最近的查询节点作为子节点的父节点，并建立查询之间的关系；(2)当mc不为0时，选取第一个拥有最大mc的点击节点作为子节点的父节点，并建立查询来自点击的关系。下面分别为mc和mmc具体求解公式：

 (3.1)

 (3.2)

其中，公式3.1中*KSC*为关键词分词集合（Keywords Split Collection），*CIC*内容索引集合（Content Index Collection），*mc*为*KSC*与*CIC*交集元素数量。公式3.2中*n*为点击节点的数量。

表3.2 时间树关系生成算法

Table. 3.2 The algorithm of generating relation between timetree nodes

|  |
| --- |
| **算法3.2** 时间树关系生成算法 |
| 算法输入：时间树*TT*,时间树新节点*N*  算法输出：时间树*TT*   1. **Begin** addTimeTreeNode(*TT*,*N*): 2. *Nc = N* 3. **If** *N* == *CN* //当新节点为点击节点 4. *Np*= getCurrentQN()//获取当前的查询节点 5. *R = CFQ //*父子节点关系为点击来自查询 6. Add( *TT.CNC*, *N*)//时间树点击节点集合添加新的节点 7. **Else** //当新节点为查询节点 8. *TT.QNC* add *N* //时间树查询节点集合添加新的节点 9. maxCount = 0 //maxCount存储关键词在文章中出现最多次数 10. **For each** *CN* in *TT.CNC* //循环获得关键词命中次数最多的点击节点 11. temp = getHitCount(*N.QK,CN*) //getHitCount 计算查询关键词在点击内容中命中次数 12. **If** temp > maxCount 13. maxCount = temp 14. *Np = N* 15. **End If** 16. **End For** 17. **If** maxCount == 0 18. *Np =* getLastQN() //获取最后一个查询节点节点 19. *R = RQQ* //父子节点关系为查询之间的关系 20. **Else** 21. *R = QFC* //父子节点关系为查询来自点击的关系 22. **End If** 23. **End If** 24. *R<Nc,Np>=*{*R,Nc,Np*} 25. add(*TT*. *R<Nc,Np>C ,R<Nc,Np>*) //时间树关系集合添加新的关系 26. **Return** *TT* 27. **End** |

本文通过算法3.1生成时间树节点，通过算法3.2将生成的时间树节点与原有时间树节点建立关系，从而完成了时间树的自动构建过程。

## 3.3基于时间树的用户搜索行为数据收集方法

时间树作为新的复杂搜索管理工具，帮助用户在复杂搜索中记录和管理查询、点击和子任务，从而减轻了用户进行复杂搜索的负担。本文研究基于时间树的搜索意图分类方法，这一研究包括三个部分：基于时间树的用户搜索行为数据收集方法的研究；基于用户搜索行为数据的用户搜索行为模型的研究；基于用户搜索行为模型的用户搜索行为分类的研究。本节首先研究基于时间树的用户搜索行为数据收集方法，以作为建立用户搜索行为模型和用户搜索行为分类的基础。

### 3.3.1基于时间树的用户搜索行为的形式化描述

为了研究基于时间树的用户搜索行为数据收集方法，首先需要定义基于时间树的用户搜索行为。下面给出基于时间树的用户搜索行为相关定义。

【定义3.10】用户搜索行为(User Search Behavior, )：基于时间树的用户搜索行为。被定义为一个五元组。其中，是进行搜索行为的用户ID(User ID)，为时间树搜索行为发生时间（Behavior Happen Time），为时间树搜索行为类型,BC为时间树搜索行为内容（Behavior Content），为时间树搜索行为目标节点ID(Target Node ID)。用户搜索行为包含以下8种类型的行为：

* 查询行为（Query Behavior，）：当用户提交了查询词进行查询，生成查询节点，视为用户进行了查询行为，目标节点ID不进行设置；
* 用户进行了查询行为（Query Behavior，）：当用户点击了查询结果页中的某一结果，生成了点击节点，视为用户进行了点击行为，目标节点ID不进行设置；
* 添加评论行为（Add Comment Behavior, ）：当用户对时间树上的节点添加评论时，视为用户进行了添加评论行为，目标节点ID为被添加评论的节点ID；
* 删除评论行为（Delete Comment Behavior，）：当用户删除时间树上的节点的某一评论时，视为用户进行了删除评论行为，目标节点ID为被删除评论的节点ID；
* 修改评分行为（Change Rate Behavior，）：当用户对时间树上的节点进行评分时，视为用户进行了修改评分行为，目标节点ID为被评分的节点ID；
* 节点拖拽行为（Drag Node Behavior，）：算法3.2给出了时间树节点关系生成算法。同时，时间树允许用户对节点进行拖拽来改变时间树节点间的关系。用户选择新的父节点，使子节点断开与原父节点的关系然后建立子节点与新的父节点的关系。子节点与新的父节点的关系仍服从定义3.7中时间树节点关系的定义。当用户对时间树节点进行拖拽时，视为用户进行了节点拖拽行为，目标节点ID为被拖动节点ID。
* 显示提示信息行为（Show ToolTip Behavior，）：当用户鼠标指向时间树节点标志持续一秒后，时间树节点提示信息由隐藏切换为显示，视为用户进行了显示提示信息行为，目标节点ID为提示信息所属的节点ID；
* 隐藏提示信息行为（Hide ToolTip Behavior，）：当用户鼠标范围脱离时间树节点提示信息时，时间树节点提示信息由显示切换为隐藏，视为用户进行了隐藏提示信息行为，目标节点ID为提示信息所属的节点ID。

综上，用户搜索行为类型可以定义为如下形式：

【定义3.11】用户搜索行为类型（Behavior Type，）：基于时间树的用户搜索行为类型。，搜索行为类型等且仅等于上述8种行为类型的任意一种。

【定义3.12】用户搜索行为集合(User Search Behavior Collection, ):用户搜索行为的集合。 ，其中为搜索行为的数量。

综上，我们给出了关于用户搜索行为的所有定义，当用户搜索行为类型BT为不同的类型时，用户搜索行为内容BC同样会有不同的取值，具体如下表所示：

表3.3 搜索行为内容取值

Table. 3.2 The value of serarch behavior content

|  |  |
| --- | --- |
| 搜索行为类型 | 用户搜索行为内容 |
| 查询行为 | 用户提交的查询关键词，同时也是生成的查询节点关键词*QK* |
| 点击行为 | 用户的点击内容标题，同时也是生成的点击节点点击内容标题*CCT* |
| 添加评论行为 | 用户添加的评论内容 |
| 删除评论行为 | 用户删除的评论内容 |
| 修改评分行为 | 评分分数 |
| 拖拽节点行为 | 由被拖拽节点的原父节点ID和现父节点ID组成的二元组 |
| 显示提示信息行为 | 空 |
| 隐藏提示信息行为 | 空 |

### 3.3.2时间树搜索行为数据收集方法

本文提供一个支持学术搜索的浏览器插件CiteXplore。CiteXplore使用时间树对搜索过程进行管理，其界面如图3.4所示。左边区域①为普通学术搜索区，右上角区域②为时间树工作区，右下角区域③为搜索任务控制台区。当用户在普通学术搜索区提交查询或进行点击，时间树工作区按照算法3.1和算法3.2构建构建时间树。

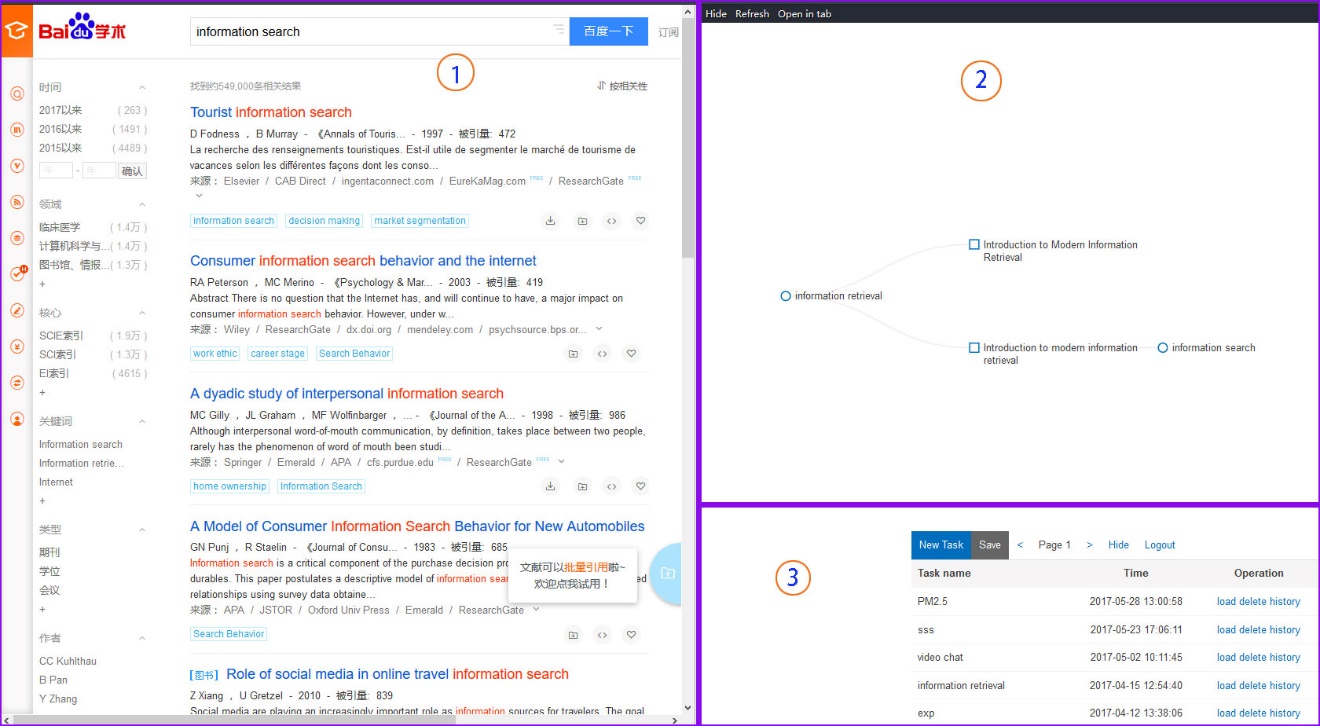


图3.4 CiteXpolre界面

Fig. 3.4 The interface of Citexplore

如图3.5所示，基于用户搜索行为所属区域不同，本文将用户搜索行为类型划分为两个大类：

1. 学术搜索区域行为类型。查询行为为用户在学术搜索区域提交查询词，同样点击行为为用户在学术搜索区域点击了某些结果页，因此查询行为和点击属于学术搜索区域行为类型。
2. 时间树区域行为类型。在定义3.11中用户的所有行为中，除查询行为和点击行为外的所有用户行为都为用户对时间树节点的操作，因此将添加评论行为、删除评论行为、修改评分行为、拖拽节点行为、显示提示信息行为、隐藏提示信息行为划分到时间树区域行为。

图3.5 用户搜索行为类型分类

Fig. 3.5 The category of User Search Behavior Types

基于搜索行为所属区域的划分，我们分别为学术搜索区域行为类型和时间树区域行为类型设计不同的数据收集方法：

1. 学术搜索区域行为类型数据收集：CiteXplore对学术搜索页面和查询结果页中所有链接进行监视。当用户提交查询时，首先解析用户查询链接并提取用户查询词，然后将搜索行为类型“查询行为类型”、查询词发送到服务器，服务器获取当前时间，然后将行为类型、查询词、当前时间封装为用户搜索行为并写入学术搜索区域用户行为日志中。同样，当用户进行了点击时，首先解析用户点击的链接并提取出文章标题，然后将搜索行为类型“点击行为”、文章标题发送到学术搜索区域服务器，服务器获取当前时间，然后将行为类型、查询词、当前时间封装为用户搜索行为并写入学术搜索区域用户行为日志中。
2. 时间树区域行为类型数据收集：CiteXplore对时间树工作区的所有时间树节点进行监视。当用户对时间树节点增加评论、删除评论、修改评分、拖拽节点、显示节点信息提示、隐藏节点信息提示时，根据表3.3中的规则获取相关用户搜索行为内容，然后将行为类型、用户搜索行为内容、被操作时间树节点ID发送到服务器，服务器获取当前时间，然后将行为类型、用户搜索行为内容、被操作时间树节点ID、当前时间封装为搜索行为并写入时间树区域用户行为日志中。

基于时间树的用户搜索行为数据收集算法如算法3.3所示：首先对普通学术搜索搜索区域（Serach Area，SA）和时间树区域（TimeTree Area，TTA）进行监视，然后获取用户操作数据并将其发送到服务器中保存到对应日志中。

表3.4 用户搜索行为数据收集算法

Table. 3.2 The algorithm of collecting user search behavior data

|  |
| --- |
| **算法3.3** 时间树行为数据收集算法 |
| 算法输入：用户的行为action  算法输出：用户行为数据   1. Begin timeTreeBehaviorCollection（） 2. If action in SA then //如果用户行为发生在普通学术搜索区域 3. ExtractTextFromUrl(action.type,action.url,&CB) //基于用户行为类型和查询或点击的url来提取查询关键词或点击的文章标题到用户行为内容中 4. SendToServer(action.type,CB,NULL) //将用户行为类型和用户行为内容发送到普通学术搜索区域服务器，时间树目标节点ID发送为空 5. Else 6. getFromTimeTree(&TTID) //从时间树中获取时间树目标节点ID 7. getFromTimeTree(action.type,TTID,&CB) //基于用户行为类型从时间树中获取用户行为内容 8. SendToServer(action.type,CB,TTID) //将用户行为类型和用户行为内容以及时间树目标节点ID发送到时间树区域服务器 9. End If 10. End |

在算法3.3中，首先判断用户的行为action是否属于普通学术搜索区域，如果用户行为action属于普通学术搜索区域，然后根据用户的提交查询的url或点击的url进行提取其中的查询关键词或点击标题。因为查询关键词或点击的标题都包含在url中，因此可以很轻易提取出查询关键词和点击的标题。最后将用户行为类型、查询关键词或点击的标题提交到服务器中如果用户性action属于时间树区域，首先从时间树中获取用户所操作的时间树节点ID，然后根据操作类型、时间树节点ID如表3.3中所示获取行为内容，最后将用户行为类型、用户行为内容以及时间树目标节点ID提交到服务器中。

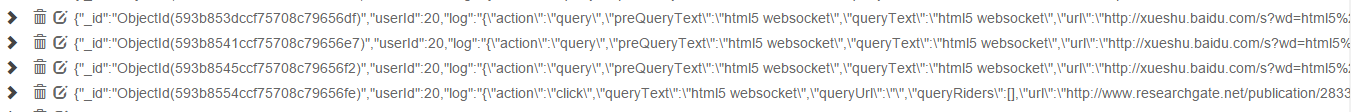
综上，本文实现了对基于时间树的用户行为数据的记录。下面分为学术搜索区域用户搜索行为日志和时间树区域用户行为日志的部分内容截图。

图3.6 学术搜索区域用户搜索行为日志

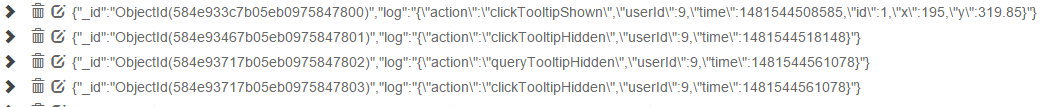
Fig. 3.6 The log of user search behavior in search area

图3.7 学术搜索区域用户搜索行为日志

Fig. 3.7 The log of user search behavior in timetree area

## 3.4 本章小结

本章主要内容描述了基于时间树的复杂搜索管理中用户行为数据的收集方法。首先，指出了现有搜索引擎不支持复杂搜素中对查询、点击和子任务的记录，然后，提出一种可视化的复杂搜索管理方法-基于时间树的复杂搜索管理方法，然后给出了时间树的具体定义、基于时间树的管理过程以及时间树的自动构建方法。基于此，提出了基于时间树的用户意图分类框架和实现过程。最后，详细介绍了基于时间树的用户搜索行为数据的收集方法。

# 第四章 基于时间树的搜索行为模型的搜索行为数据预处理方法

本章主要描述基于时间树的搜索行为模型的搜索行为数据预处理方法。首先，详细描述搜索行为模型的定义。其次，针对搜索行为模型中搜索行为数据获取的问题，本文详细描述了基于时间树的复杂搜索任务的设计和实验过程。最后对用户的实验数据进行分析并建立搜索行为模型，并搜索行为模型给出搜索行为数据预处理方法。

## 4.1 基于时间树搜索行为模型定义

由于用户搜索行为的不确定性，用户搜索行为数据中存在大量噪声数据，为了有效的行为分类，首先需要对用户搜索行为数据进行预处理。为了进行数据预处理，本文首先建立基于时间树搜索行为模型。下面给出搜索行为模型的形式化描述。

【定义4.4】用户搜索管理行为(User Search Manager Behavior)： 用户对搜索过程进行管理的行为。发生在时间树工作区的一切用户行为都为用户对搜索过程进行管理的行为。在图3.5中，时间树工作行为包括添加评论行为、删除评论行为、修改评分行为、拖拽节点行为、显示节点提示信息、隐藏节点提示信息。然而，显示节点提示信息和隐藏时间树节点信息都是成对出现的。因此可以认为显示节点提示信息和隐藏节点信息的目的是为了展开节点提示信息。其次，添加评论行为、删除评论行为、评分行为都需要在节点提示信息中进行。因此，对于显示节点信息和隐藏节点信息中不出现添加评论、删除评论、修改评分的行为，认为用户在回顾搜索内容；否则认为用户进行了评论、删除评论、修改评分的行为。因此管理搜索行为表示为。其中ACB、DCB、CRB、DNB分别为添加评论行为、删除评论行为、修改评分行为、拖动节点行为，RB为回顾行为(Review Behavior)。

【定义4.1】搜索行为模型（Search Behavior Model）：用来描述用户搜索行为的模型。搜索行为模型为一个，SBM = {QBC,CBC,USMB ,R}。其中，QBC={QB1，QB2,…QBn}表示查询行为集合，CBC={CB1,CB2,…,CBn}表示点击行为集合，TABC为时间树区域行为集合(TimeTree Area Behavior Collection)，R表示各种行为之间的关系。

用户行为模型中的行为模型关系包含两个部分：

1）用户查询行为与用户搜索管理行为中各个行为之间的关系，本文简称为查询行为与管理行为之间的关系。查询行为和管理行为之间的关系又可划分为查询行为与后续管理行为之间的关系和管理行为与后续查询行为之间的关系。

2）用户点击行为与用户搜索管理行为中各个行为之间的关系，本文简称为点击行为与管理行为之间的关系。点击行为和管理行为之间的关系又可划分为点击行为与后续管理行为之间的关系和管理行为与后续点击行为之间的关系。

在复杂搜索中，存在两种典型的复杂搜索任务：当用户搜索任务明确，但是缺少相关搜索任务领域知识，用户需要进行大量的查询和点击从而不断的学习和补充相关领域知识，最终完成搜索任务，例如对机器学习不太了解的用户想要撰写一篇深度学习领域的综述，此类型的复杂搜索任务称之为学习型任务；当用户搜索任务目标不明确时，用户通过试探性的查询和点击不断的来调整自己的搜索方向，从而完成搜索任务，此类型的搜索任务称之为探索型任务。基于复杂搜索任务类型的划分，用户行为模型划分为两个模型：学习型任务搜索行为模型和探索型任务搜索行为模型。

## 4.2复杂搜索实验

上述我们给出了用户搜索行为模型的定义，为建立用户搜索行为模型，本文首先需要对用户搜索行为数据进行收集。在基于时间树的用户搜索行为数据收集方法的基础上，本文设计并进行复杂搜索实验，从而完成对用户搜索行为数据的收集，进而对用户行为进行分析并完成用户搜索行为模型。

### 4.2.1复杂搜索实验设计

为建立学习型任务搜索行为模型和探索性任务行为模型，需要收集学习型任务搜索行为数据和探索型任务行为数据。因此，本文分别设计了学习型搜索任务和探索型搜索任务。为保证实验结果不受限于实验内容，本文分别设计两个学习型搜索任务和两个探索型搜索任务。下面分别4个任务的具体细节和要求。

【任务一】化疗领域知识搜索：搜索相关论文，了解化疗中的常用药物，药物机理，使用方法，适用情况及副作用等，尤其针对癌症治疗，了解化疗药物的治疗手段，组合方式等。（不仅仅是任务背景中提到的支持化学疗法的药物，还要求了解对化学疗法进行辅助的药物，如缓解化学疗法副作用的药物等。

【任务二】PM2.5领域知识搜索：了解PM2.5 的概念、来源；了解PM2.5 的危害以及危害原因；找到中国PM2.5 污染最为严重的几个城市，并分析其形成原因；研究降低PM2.5 的方法，并分析其原理以及其实际实施可能性（包括但不局限于利弊）。

【任务三】给出S. Shunmuga Krishnan, Ramesh K. Sitaraman的文章Video Stream Quality Impacts Viewer Behavior: Inferring Causality Using Quasi-Experimental Designs的摘要部分，要求实验参与者搜索并给出与摘要内容有关的文章标题。

【任务四】给出Hanqiang Cheng, Yu-Li Liang, Xinyu Xing, Xue Liu, Richard Han, Qin Lv, Shivakant Mishra文章Efficient misbehaving user detection in online video chat services的摘要部分，要求实验者参与者搜索并给出与摘要内容有关的文章标题。

其中，任务一和任务二属于学习性任务，学习性任务要求实验参与者给出搜索结果报告，搜索结果报告尽可能给出任务要求答案。任务三和任务四属于探索型任务，探索型任务要求实验参与者给出和摘要内容有关的相关文章标题。

### 4.2.2 复杂搜索实验过程

用户在使用搜索引擎进行复杂搜索时，需要进行大量的查询和点击。这些查询和点击需要占用用户大量的时间，使得用户的搜索任务分为多次进行完成。为模拟复杂搜索用户搜索任务分多次完成的情况，任务一、任务二设计分为3次进行搜索，每次搜索25分钟；任务三、任务四设计分为2次进行搜索，每次搜索25分钟。

为保证实验参与者们有着相近的领域知识和搜索经验，我们募集了16名东北大学硕士研究生进行实验。其中，硕士3年级学生1名，硕士2年级学生7名，硕士1年级学生8名。实验采用火狐浏览器开发版作为搜索浏览器，采用CiteXplore作为用户行为收集工具，使用百度学术搜索引擎作为唯一搜索引擎。

在第一次实验开始前，我们首先对实验参与者进行统一培训，使实验参与者熟悉时间树对搜索过程管理的方式和时间树的相关操作。并且，给实验参与者每人留出15分钟试用搜索环境，使实验参与者搜索并使用时间树进行管理其感兴趣的内容。在实验参与者试用期间，实验人员指出参与者使用时间树进行管理搜索任务的不当之处并给相关建议。下面正式开始实验，实验过程分三次进行，具体过程如下：

1. 第一次实验：首先用户抛掷一次硬币，根据硬币的正反来确定搜索任务一还是搜索任务二。同样，用户搜索任务三还是任务四也是通过抛掷硬币的方式进行得出。为保证实验样本的均匀，实验要求每个实验都有半数人进行。当其中一个任务被选择次数已经达到8次，剩余未抛掷硬币的实验者自动选择同一个任务类型下的另一个任务。此时，所有的实验参与者都已经选择出自己需要进行的搜索任务。其次，任务三和任务四要求用户搜索与摘要内容有关的相关文章。为使得任务三和任务四符合探索型任务的要求，需要使得用户的搜索目标不明确，因此在实验开始时，给实验者留出10分钟的时间去阅读探索型任务摘要内容，但并不告诉实验参与者阅读此摘要的原因，此后实验参与者不能再去查看此摘要。待实验参与者看完摘要后，给出实验参与者学习型任务的任务要求，告诉其实验分3次完成，每次25分钟，每次并留出25分钟时间进行搜索，搜索期间时间参与者可随时查看任务要求。
2. 第二次实验：第二次实验在第一次实验开始后的48小时后开始。首先，告诉实验参与者搜索和第一次实验看的英文摘要内容有关的文章并说明此任务分2次完成以及每次搜索的时间，并留出25分钟的时间进行搜索。搜索完毕后，实验人员帮助实验参与者加载出第一次实验后形成学习型任务的时间树，并在此基础上进行25分钟的搜索。
3. 第三次实验：第三次实验在第二次实验开始后的48小时后开始。首先，帮助实验参与者加载其在第二次实验后形成的探索型任务的时间树，并基于此时间树留出相应的时间给实验参与者进行搜索。待其搜索完毕后，再帮助参与者加载出第二次实验后形成的学习型任务的时间树，并进行搜索。

第三次实验结束后的24小时后，邀请实验参与者撰写学习型任务的实验报告以及给出与探索型任务的相关文章标题。在此期间，实验参与者可以对通过自己搜索形成的时间树和下载的文章进行访问。

## 4.3基于时间树的用户搜索行为模型的建立

通过复杂搜索实验，基于时间树的用户搜索行为数据收集方法，完成了对用户搜索行为数据的收集。在4.1节中，给出了基于时间树的用户搜索行为模型的定义。本节通过对用户搜索行为的分析建立基于时间树的用户搜索行为模型。

### 4.3.1基于时间树的用户搜索行为分析

为建立基于时间树的用户搜索行为模型，首先对用户搜索行为数据进行分析。根据定义4.2，用户搜索行为模型包括用户的查询行为、用户的点击行为、用户的搜索管理行为以及查询行为与管理行为的关系、点击行为与管理行为的关系构成。因此，基于时间树的用户搜索行为分析将主要根据这几个方面进行展开。

本文首先对用户在使用时间树中的各种行为进行定量分析，基于任务类型的分类统计出各个行为的数量。

图4.1 全部实验参与者学习型任务和探索型任务的查询数量和和点击数量和柱状图

如图4.1是全部实验者的学习型任务和探索型任务的查询数量和点击数量。从图中可以明显观察到，对于学习型任务，点击数量远远超过查询数量，点击数量为387，查询数量为207，点击数量大约占查询数量和点击数量总和的百分之65，而查询数量只占到总数量的百分之35左右。但是，对于探索型任务，查询数量的总和点击数量的总和是差不多的。查询数量为180，点击数量为215，查询数量占查询和点击总数量的百分之45，点击数量占总数量的百分55。为避免学习型任务和点击型任务的查询和点击的比例受到任务不同的影响。下对每个任务的查询数量和点击数量的进行统计。

图4.2 全部实验参与者四个任务的查询数量和和点击数量比例柱状图

如图4.2是全部实验参与者四个任务的查询数量和点击数量比例柱状图，从图中可以观察到，任务1和任务2的查询数量占查询和点击的总数量的都在百分之35左右，任务3和任务4的查询数量占查询和点击总数量的百分之45左右。因此任务的的查询数量和点击数量比例主要受到任务类型的影响较重，而几乎不受到任务具体内容的影响。

为了解任务的查询数量和点击数量受任务类型的影响的原因，对用户的所有查询图图4.3

Fig 4.3

的点击次数进行统计，如图3.3分别是学习型任务和探索型任务的查询后的点击数量频率饼图。从图上可以看出，探索型任务有百分之47的查询后没有点击，而对于学习型任务只有百分29的查询后没有点击。而且对于一个查询后接有两个及两个以上的点击，学习型任务的比例是远远超过探索型任务的。观看实验参与者的实验录像以及对实验参与者的后续调查后发现，对于探索型任务，实验参与者由于只记得摘要中的大概讲的内容，因此实验参与者需要提交大量查询词来逐渐逼近搜索目标。在提交查询词后，实验参与者可以通过两种方式来确定自己查询的主题是否贴近于摘要中的内容：通过查看查询结果页中的点击的标题，这种情况下不需要进行点击；或者通过点击然后查看点击中的文章来确定自己的查询是否贴切想要搜索的内容。在探索型任务中，实验参与者有着更多的重构，重构后搜索引擎返回结果往往和重构前的结果是一样的，因此在这种情况实验参与者也无需进行点击来确定自己的查询是否贴近目标。对于学习型任务，由于实验参与者有着明确的任务需求，因此，他们有着明确的搜索目标。然而在提交查询词后，实验参与者需要进行点击来获取能够满足任务要求的知识。因此，在学习型任务中实验参与者查询后不进行点击的几率较探索型任务较少。从比重上来看，学习型任务百分之29后查询不进行点击在整个饼图中也是占有最大的比例，从实验录像中分析，这是由两点原因造成的：一是有些查询词不能直接得到实验参与者所需要的结果，因此他们需要进行重构来在此尝试获取相关内容；二是由于当实验参与者完成一个子任务后，有时会再进行尝试搜索一下来确定是否还有内容进行补充。

上述统计了所有的查询和点击行为，下面首先对两种任务的时间树管理行为出现的次数进行统计。如表4.1所示是用户搜索管理行为出现的次数。

表4.1 各种用户搜索管理行为出现次数

Table 4.1 The count of each user search behavior

|  |  |  |
| --- | --- | --- |
| 用户搜索管理行为 | 学习型任务 | 探索型任务 |
| 增加评论行为（ACB） | 91 | 3 |
| 删除评论行为（DCB） | 0 | 0 |
| 修改评分行为(CRB) | 49 | 18 |
| 拖拽节点行为(DNB) | 62 | 46 |
| 回顾(RB) | 361 | 214 |

从表中可以观察到，在学习型任务中和探索性任务中，都没有实验参与者进行删除评论行为。此外，可以很明显从表中观察出的一种情况：学习型任务的添加评论行为远远大于探索型任务的添加评论行为。通过对实验参与者的实验后调查和对任务的分析得知：在学习型任务中，要求撰写报告，因此实验者需要把写报告有关的内容写到评论中，此外，对于支持实验者进行下一步搜索的信息，实验者也往往把他们记录在评论中；而对于探索型任务，只要求给出相关文章标题，因此实验参与者只需将点击标记为是否为相关文章即可，而无需记录文章中的内容，这一点使用评分功能即可达到。

图4.4

Fig4.4

由于任务的复杂性不同，学习型任务进行了3次实验，探索型任务进行了2次实验，因此求得每次实验的平均用户搜索管理行为对于对比两种任务行为的不同是有价值的。如图4.4是两类任务的用户平均每次实验的搜索管理行为数量柱状图。从中可以看出，两类任务的拖拽节点的次数和回顾的次数都是几乎差不多的。如图4.5两类任务中评分的频率柱状图。对于评分行为的差异，通过对实验者进行评分分值以及评分对象进行统计，发现用户只有对点击节点有评分行。此外，用户有着更多的打4分和打5分的行为，而打1分和2分以及打3分的行为很少。相对于学习型任务，实验参与者更加难以找到所需要的文章，因此造成了探索型任务评分数量少于学习型任务的评分数量。

图4.5

Fig. 4.5

为建立基于时间树的用户搜索行为模型，本文基于用户的的搜索行为日志，将用户搜索行为日志按照发生时间进行排序后，本文将紧邻的两个行为之间的跳转成为两种行为之间的关系。下面首先对用户搜索行为的关系进行统计分析。

图4.6 学习型任务的各种行为跳转数量柱状图

图4.7 探索型任务的各种行为跳转数量柱状图

Fig. 4.7

图4.6为学习型任务的各种搜索行为跳转数量图,图4.7为探索型任务的各种搜索行为跳转数量图。下面本文分别从查询行为、点击行为、添加评论行为、修改评分行为、拖拽节点行为、回顾行为的跳转去向进行分析讨论：

查询行为跳转去向：学习型任务的查询行为后最多的行为为点击行为，而且远胜于其他行为。而在探索型任务中，查询行为后最多的行为为查询行为。分析用户搜索日志和任务要求，在学习型任务中，用户任务需求比较明确，用户产生的最多的操作是查询后获取相关点击进行查询；而在探索型任务中，用户由于需求不明确而经常需要切换查询词来尝试获得正确的搜索方向。除了跳转到查询行为和点击行为外，查询后用户进行拖拽节点来组织时间树的行为也占有一部分比重。然而用户进行查询后很少进行评论节点行为，这与之前用户很少对查询节点进行评论的结论是一致的。

点击行为跳转去向：查询和点击作为构成搜索过程最基础的元素，无论在学习型任务中还是在探索型任务中，点击后用户最多的行为为查询和点击。通过回顾实验者的学习型任务的搜索过程视频，发现在用户搜索过程中用户进行点击和查看相关文章后，出现许多对点击进行评论的行为。但是，在图4.6中，实验参与者点击后进行评论的行为只出现了3次。通过对用户的调查后，实验参与者在点击后想要去评论，但是显时提示信息后，忘记去评论什么，因此又隐藏起来，从而造成了一次回顾行为而没有进行评论行为。但是在之后，实验者通过再次看文章的方式又重新进行了评论，但此时的用户行为被归为了回顾行为跳转到点击的行为。此外，在学习型任务和探索型任务中，都出现了少量的点击行为跳转到拖拽节点的情况。此时，用户所进行拖拽的情况分为两种：一是用户对当前点击节点进行了拖拽，但这种行为在时间树中是不合理行为，因为根据时间树节点的生成规则，点击节点的位置跟在当前查询词之后是没有问题的，因此无需对点击节点进行拖拽行为；二是用户对其他节点进行了拖拽行为，这种情况可能是一种合理行为。下面将对点击后的拖拽进行统计和分析。

添加评论行为跳转去向：在探索型任务中，实验者进行添加评论行为次数过少，因此此处只讨论学习型任务中添加评论行为的跳转。首先，添加评论后实验者后续最常见的行为为回顾行为。此外，添加评论后继续进行评论也有不少的数量，这是由于某些点击中文章含有多条有用信息，因此，实验参与者需要添加多条评论。实验参与者在对节点进行评论后会对节点进行评分功能，对节点进行评分后，节点标志会自动变色，可以更加容易的找到被评分的节点。然后，在其中有少量用户添加评论行为后进行了查询行为。用户此时进行的查询可以分为两种，第一种是用户根据添加评论的内容进行了后续搜索，第二种是用户后续查询与添加评论的内容无关。下面对这两种行为进行统计分析。

修改评分行为跳转去向：在学习型任务中，修改评分行为后有5种行为，分别为添加评论行为、点击行为、查询行为、修改评分行为、进行回顾行为。在探索型任务中， 由于添加评论行为过少，因此在修改评分行为后只有点击行为、查询行为、修改评分行为、进行回顾行为等四种行为。对于用户修改评分行为，可以简单的认为分为两种：第一种是对节点进行了好的评价（评4分或评5分），第二种是对点击进行了中等及以下评价（评3分及以下）。为解释用户评论后的点击，下面对实验者评分后进行评论的评分分值进行统计。

图4.

Fig.

图4.是实验者修改评分后跳转到添加评论行为时5种评分的频数。从图中看出，4分和5分各出现了5次，3分出现了1次，1分和2分没有出现。当实验者对某个节点进行打高分时，认为此节点对完成任务有帮助或可以支持用户进行下一步搜索，因此需要进行添加评论来记录此节点中的信息。

在图4.6和4.7中出现了用户在评分之后继续进行评分的行为。用户连续进行修改评分有两种情况：第一种是实验参与者对之前的评分不满意，想修改一下评分；第二种是实验参与者同时对个节点进行评分。下面是对这两种情况的统计数据。

图4.

Fig.

图4.

Fig.

图4.是连续两次修改评分行为是否是相同节点的频数柱状图。如图在所有连续修改评分行为中，有一半多是对同一节点进行评分，有一少半是不同的节点进行评分。对于进行连续对同一节点进行评分的行为，可认为其只对节点进行了一次评分且评分分值为最后一次评分的分值。对于不同节点进行连续评分的行为，则可视为之前忘记进行评分，现在对评分进行补充。

此外，在图4.6和图4.7中，修改评分行为跳转后的行为后最多的行为是回顾行为。回顾行为是指用户显示了提示信息后并未进行其他操作的行为。然而修改评分操作也需要在提示信息中进行修改。因此，在用户修改评分后可能有出现误操作而引发了回顾行为。本文判断修改评分行为中的节点和后续回顾行为中的节点是否是同一节点来判断用户行为是否是误操作。图4.13中，学习型任务和探索型任务中，修改评分行为的节点和之后回顾的节点是相同的次数大约为不同的次数的3倍。

拖拽节点行为跳转去向：在学习型任务和探索性任务中，拖拽节点行为后有查询行为、点击行为、回顾行为以及拖拽节点行为。前文中已经分析，拖拽节点行为的产生主要在查询行为之后，再查询行为后，用户最常见的行为为查询行为和点击行为。而拖拽节点行为是用户查询节点与父节点关系生成不准确而产生的行为。因此，在拖拽节点行为后，用户产生的行为与查询行为后产生的行为基本一致。

回顾行为跳转去向：在学习型任务和探索型任务中，回顾行为是所有行为中最多的一种行为，回顾行为后有查询行为、点击行为、添加评论行为、修改评分行为、拖拽节点行为、回顾行为。使用时间树进行搜索过程管理时，很多操作会在tooltip中进行，例如例如添加评论、修改评分等。因此可能会产生许多回顾的误操作。由于查询和点击行为不直接对时间树进行操作，因此查询行为和点击行为不会对产生误操作。

图4.

Fig.

图4.

Fig

图4.为回顾后修改后的

表4.1

Table 4.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 跳转前行为 | 跳转后行为 | 学习型任务中的数量 | 探索型任务中的数量 |
| 1 | 查询行为（QB） | 点击行为(CB) | 115 | 81 |
| 2 | 查询行为（QB） | 查询行为(QB) | 51 | 87 |
| 3 | 点击行为（CB） | 点击行为（CB） | 228 | 118 |
| 4 | 点击行为（CB） | 查询行为（QB） | 111 | 76 |
| 5 | 添加评论行为（ACB） | 添加评论行为(ACB) | 17 | 0 |
| 6 | 添加评论行为（ACB） | 拖拽节点行为（DNB） | 10 | 0 |
| 7 | 添加评论行为（ACB） | 回顾行为（RB） | 49 | 3 |
| 8 | 查询行为（QB） | 回顾行为（RB） | 22 | 29 |
| 9 | 查询行为（QB） | 拖拽节点行为（DNB） | 37 | 31 |
| 10 | 拖拽节点行为（DNB） | 点击行为（DB） | 37 | 24 |
| 11 | 修改评分行为（CRB） | 添加评论行为(ACB) | 11 | 0 |
| 12 | 修改评分行为（CRB） | 回顾行为（RB） | 23 | 8 |
| 13 | 回顾行为（RB） | 添加评论行为（ACB） | 60 | 3 |
| 14 | 回顾行为（RB） | 点击行为（CB） | 26 | 25 |
| 16 | 回顾行为（RB） | 查询行为(QB) | 38 | 39 |
| 17 | 回顾行为（RB） | 修改评分行为（CRB） | 24 | 16 |
| 18 | 回顾行为（RB） | 回顾行为（RB） | 186 | 116 |

表4.1为学习型任务和探索型任务各种搜索行为跳转数量统计表。其中，查询和点击作为搜索过程最基本的组成元素， 因此，查询之后进行查询、查询之后进行点击、点击之后进行查询、点击之后进行点击这四种行为跳为是经常出现的跳转情况。为建立用户搜索行为模型，本文将上表中一些行为跳转情况进行合并或删除，具体规则和依据如下：

1. 时间树上的行为自我跳转情况：时间树上的行为的自我跳转主要包括两种情况：添加评论跳转到添加评论行为、回顾行为跳转到回顾行为。当用户进行的点击或查询后包获得多种收益后，时间树允许用户对时间树上的节点进行多次评论。因此，添加评论行为跳转到添加评论可以划分为两种情况：用户对时间树上的同一节点进行添加评论；用户对时间树上的不同节点进行评论。因此，本文首先对添加评论行为的跳转进行合并。当用户连续对时间树上同一节点进行添加评论时，可以视为是一次添加评论行为，而不是添加评论行为跳转。因此，首先进行统计连续进行评论时评论的节点是同一个节点的数量来判断合并添加评论行为跳转到添加评论行为的有效性。如图4.10中a所示，所有连续评论行为都为对同一个节点进行评论。因此，可以全部添加评论行为跳转到添加评论行为的情况合并为一次添加评论行为。而回顾行为和添加评论行为类似，当用户进行连续回顾时，所回顾的节点可能为同一节点。连续对同一节点进行回顾时可以认为是一次回顾行为。因此，可以将连续回顾同一节点的行为合并为一次回顾行为。为评价此合并行为带来的收益，本文对两类任务中连续回顾行为是否是同一节点的数量进行了统计。图4.10中b为学习型任务和探索型任务中连续回顾行为中的节点是否相同的数量柱状图，可以看出，节点相同的数量虽然低于不同的数量，但仍占有很大一定的数量，因此合并这种行为对无效的回顾跳转。

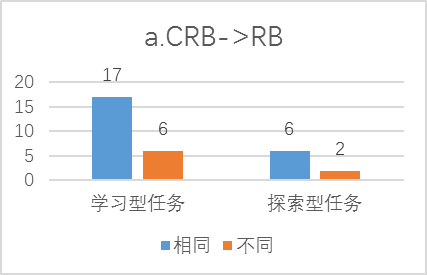
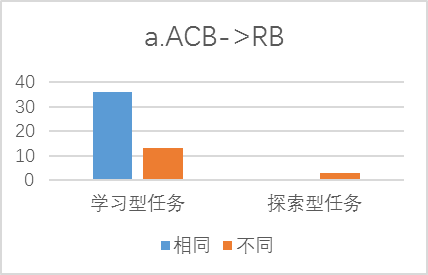
图4.10

Fig. 4.10

1. 其他时间树搜索行为跳转到回顾行为：这里指代的其他时间树搜索行为为添加评论行为、修改评分行为。当用户进行添加评论行为、修改评分行为以，都跳转到回顾行为。如被添加评论或被修改评分的节点与回顾的节点是同一个节点，可以认为此回顾行为是误操作行为，因此，可以将添加评论行为、修改评分行为跳转到回顾行为合并成添加评论行为、修改评分行为。图4.11 a和b分别为添加评论行为跳转到回顾行为、修改评分行为跳转到回顾行为中节点相同与不同的数量。从图中，可以看出，出现其他时间树搜搜行为跳转到回顾行为中的节点数量不同的情况是非常少，添加评论后和修改评分行为后进行回顾不是一种典型的行为。

图4.10

Fig. 4.10



1. 回顾行为跳转到其他时间树搜索行为：这里指代的其他时间树搜索行为添加评论行为、修改评分行为。同样，当用户想要进行上述两种行为时，有可能在之前误触发了回顾行为。本文针对回顾行为与后续添加评论行为、修改评分行为所操作的节点是同一个节点的情况视为只进行了添加评论行为和修改评论行为。图4.10表明在大多数回顾行为与后续的添加评论行为和修改评分行为所操作的节点都为同一个节点。

图4.10

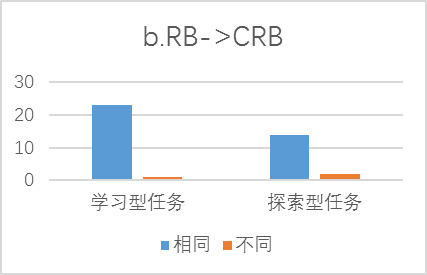
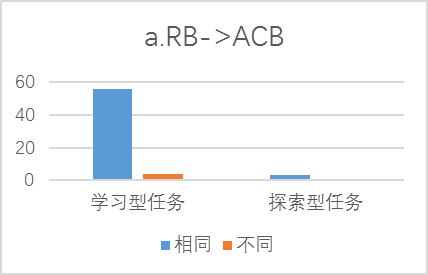


Fig. 4.10

综上，本文给出了所有基于时间树的用户搜索行为跳转的合并与删除规则，如表4.2所示：

|  |  |  |
| --- | --- | --- |
| 跳转前行为 | 跳转后行为 | 合并或删除规则 |
| 添加评论行为 | 添加评论行为 | 两次行为操作节点相同时，合并为一次添加评论行为 |
| 添加评论行为 | 回顾行为 | 两次行为操作节点相同时，删除回顾行为 |
| 修改评分行为 | 回顾行为 | 两次行为操作节点相同时，删除回顾行为 |
| 回顾行为 | 添加评论行为 | 两次行为操作节点相同时，删除回顾行为 |
| 回顾行为 | 修改评分行为 | 两次行为操作节点相同时，删除回顾行为 |

除上述跳转关系的合并和删除之外，本文将添加评论行为与修改评分行为进行合并。添加评论行为与修改评分行为互相之间有多次的跳转，而且进行评论的节点与修改评分的节点为同一个节点。用户进行修改评高分和添加评论行为都体现了节点的重要性。由于用户大多数评分行为都为评高分行为，因此将添加评论行为和修改评分行为进行合并，称为评论评分行为（Comment And Rate Behavior，CARB）。

基于上述跳转关系的合并和删除以及跳转后规则，对表4.1中用户跳转行为进行合并或删除，得到跳转行为如表4.3所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 跳转前行为 | 跳转后行为 | 学习型任务中的数量 | 探索型任务中的数量 |
| 1 | 点击行为 | 点击行为 | 228 | 118 |
| 2 | 查询行为 | 点击行为 | 115 | 81 |
| 3 | 点击行为 | 查询行为 | 111 | 76 |
| 4 | 回顾行为 | 回顾行为 | 61 | 77 |
| 5 | 查询行为 | 查询行为 | 51 | 87 |
| 6 |  |  |  |  |