**评审专家匹配算法Benchmark方案**

*2021/04/10 V1.0*

**1 问题定义**

评审专家匹配问题的定义为：针对指定的结构化学术文献，从预先指定的专家库中推荐出研究领域与学术文献描述内容所表达的研究领域最为匹配的若干候选专家，并按照匹配度由高到低排序。其中，学术文献和专家信息包含以下结构化信息：

**学术文献信息**

1. ID：唯一标识文献的标识符。
2. 标题：文档的描述性标题。
3. 摘要：描述文献研究内容、方法、结果等内容的文本，通常为一段自然语言内容。
4. 关键词列表：文献作者人工提供的4-5个关键词（组）。

**专家信息：**

1. ID：唯一标识专家的标识符。
2. 姓名：专家姓名。
3. 研究领域：专家自己提供的若干描述其研究领域的关键词（组）。
4. 学术指标：H指数，论文发表数量，总引用数等指标。
5. 研究成果：专家历史上发表的若干论文或科研项目，每条成果信息均包含标题、摘要、引用量、发表年份等。

**2 评测数据集**

**2.1 论文评审数据集**

**论文数据集**

采用论文数据作为待评审学术文献数据，其每条论文信息包含前述学术文献信息的所有结构化字段数据。此外，论文的引用文献列表将用于构造候选专家库。

**候选专家库**

构建方法如下。

对每篇待评审论文Pi，执行如下操作：

1. 提取引用文献列表，将引用文献的全部作者集合Mi加入候选专家列表；
2. 统计上述作者的出现频次，将频次大于N（N>=2）的作者标记为专家集Ri；
3. 加入随机干扰用专家若干。

最后，将所有专家集Ri和随机干扰用专家混在一起构成候选专家数据集。

**结果集**

构造遵循如下规则：

对于每篇论文Pi：

1. 专家集Ri作为“正确”评审专家列表，并按照上述频次由高到低排序；
2. 频次相同时，依次按照H指数和总引用数排序。

**2.2 项目评审数据集**

**项目数据集**

采用项目申请或报告作为待评审学术文献数据，其每条项目信息包含前述学术文献信息的所有结构化字段数据。此外，项目的承担人和引用文献列表将用于构造候选专家库。

**候选专家库，目前采用方法二**

**构建方法一**

对每个待评审项目Pi，执行如下操作：

1. 提取项目承担人Li，将其加入候选专家列表；
2. 提取引用文献列表，将引用文献的全部作者集合Mi加入候选专家列表；
3. 统计上述作者的出现频次，将频次大于N（N>=2）的作者标记为专家集Ri；
4. 加入随机干扰用专家若干。

**构建方法二**

1. 将待评审项目根据其内容进行聚类（例如关键词相同等）；
2. 对每个类别Ci，其中除项目Pi的原承担人外的其他承担人标记为集合Ri。

**结果集**的构造遵循如下规则：

对于每个项目Pi：

1. 项目原承担人Li作为“正确”评审专家1；
2. 专家集Ri作为“正确”评审专家列表2，按照上述频次由高到低排序；
3. 频次相同时，依次按照H指数和总引用数排序。

**2.3 委内评价接口及评价方法**

**基本要求**

项目组可以通过委内评价接口获知一批申请书推荐专家的评价结果，但无法得知项目主任每篇申请书具体选择了哪几位专家（下文以所有申请书都选择5位专家为例）。

**评测接口格式**

共2个评测接口，测试接口 /test 与正式评测接口 /evaluate，两个接口的输入输出数据格式为json，且形式完全一致。

输入数据包含项目组某次推荐专家的结果，数据中包含多篇申请书，每篇申请书提供申请书id以及对应的推荐专家。

输入样例：

{

"recommend":[

{

"project\_id": "12037101",

"experts:[

"12321", "21300", "12847", ...

]

},

{

"project\_id": "2819312",

"experts:[

"12321", "21300", "12847", ...

]

}

]

}

输出数据为测试数据集中，每篇申请书推荐结果的 MAP 值，计算方法见后文 评价方法 部分。

输出样例：

{

"success": 1,

"results":[

{ "project\_id":"12037101", "MAP": 0.172 },

{ "project\_id":"2819312", "MAP": 0.128 },

...

]

}

**评测接口构建方式**

项目组提供一个简单后端工程文件（可通过容器进行包装）给信息中心，代码由委内技术专家审阅（如宫辰老师），这一后端工程代码量预期在100行左右，因此可以进行白盒检查。

该后端工程依赖三个文件，文件1 test\_dataset.csv 为 csv 格式，内部数据每一行是一篇申请书及对应的真实评测专家，格式如下例：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 申请书id | 专家1 | 专家2 | 专家3 | 专家4 | 专家5 |
| 10000001 | 12300 | 12301 | 12302 | 12303 | 12309 |
| 10000002 | 19830 | 11293 | 18471 | 33928 | 12993 |
| …… | | | | | |

项目组提供样例 test\_dataset.csv ，后续由信息中心老师更换为真实数据。

文件2 evaluate\_dataset.csv 为正式评测用数据集，其格式与 test\_dataset.csv 相同。

文件3 config.txt 为配置文件，仅包含一个字符，为 0 或 1，当配置中的字符为 1 时，正式评测接口不可用。

**评测接口使用方法**

上述评测后端工程文件由信息中心老师部署至某台服务器上，并开放 /test /evaluate 两个接口给项目组。项目组对该服务器无其他访问权限。config.txt 文件中的设置默认为 0 。

当项目组需要进行结果评测时，调用 /test 接口，可以得知某次推荐的评价指标。

当项目需要进行正式评价时，config.txt 文件中的设置变为 1，项目组获得一次正式评价指标。随后将 config.txt 中的设置变为 0。

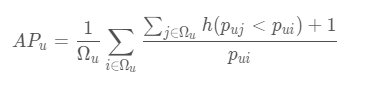
**评测接口的具体使用限制**

1. 评测工程中限制 /test 接口每天调用次数上限为 10 次，/evaluate 接口每天调用上限为 2 次。超出次数则在当天内固定返回评测失败，不计算评测值。
2. 调用接口每次上传的申请书必须包含评测集中的全部申请书，否则返回评测失败，不计算评测值。
3. 调用接口每次上传的申请书的所有推荐专家必须满 200 人，否则返回评测失败，不计算评测值。

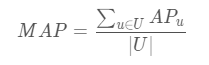
上述限制可以避免通过反复调用接口，或构造特殊数据的方式获知真实申请书推荐专家结果。

**评价方法**

采用MAP（Mean Average Precision）指标评价评审专家匹配算法结果的精度结果。对于文献u的评审专家匹配结果列表的AP值：



MAP值：



特别的，对于项目“正确”评审专家1，单独计算其在匹配结果中的命中率。