基于相关参考文献和出版物权重的自动审稿人分配

Tamim Al Mahmud*, B M Mainul Hossain †, Dilshad Ara ‡
Department of Computer Science and Engineering*†
Green University of Bangladesh, Dhaka-1207, Bangladesh*
University of Dhaka, Dhaka-1000, Bangladesh †
Dhaka International University, Dhaka-1205, Bangladesh‡
Email: tamim@cse.green.edu.bd*, mainul@iit.du.ac.bd †, dilshadara1995@gmail.com‡

摘要—每天可能有大量研究文档提交给会议、文集、期刊、通讯、年报、日报和各种定期刊物。许多这样的出版物使用无偏见的外部专家进行评估。这种做法通常称为同行评审，而这些评估者被称为审稿人。但并不总是能够挑选出最适合审查的最佳审稿人。此外，新的研究领域在各个部门不断涌现，研究数量也在急剧增加。然而，为了审查所有这些论文，每个期刊都指派了一个小型的审稿团队，他们可能并不是所有领域的专家。例如，在通信技术领域的研究论文应由同一领域的专家进行评审。因此，高效地选择最佳审稿人或审稿者对一篇研究论文来说是一个巨大的挑战。在这项研究工作中，我们提出并实现了一种程序或软件，我们在其中使用了新的策略来自动选出最佳审稿人。众所周知，每篇研究论文的末尾都包含参考文献，并且这些参考文献通常与论文来自同一领域。在此工作过程中，首先收集参考文献，并计算那些至少有一篇论文属于参考部分的作者数量。一旦收集到作者的名字，就自动浏览网络以提取研究主题关键词。然后再次搜索寻找特定领域的顶尖研究人员并计算他们的 h 指数、i10 指数和前 n 位作者的引用次数。之后，根据排名分数对前 n 位作者进行排序，并自动浏览他们个人主页以检索电子邮件地址。我们还会从网上检查他们的合著者和同事，并相应地将它们从名单中删除。剩下的前 n 位作者（研究人员）通常是教授，可能是审查研究论文的最佳审稿人。

自动频率查找算法（*AFFA*）、自动引用查找算法（*ACFA*）、自动排名算法（*ARA*）和自动电子邮件查找算法（*AEFA*）

I. 介绍

发表文章是学者专业生活的重要组成部分。尽管，写作不是每个人喜欢的活动，并且完成并发布一篇文章可能会是一个单调和缓慢的过程 [1]。幸运的是，通过采取一些自信的策略和观察，选择最佳审稿人（同行评审）这一重要障碍可以使复制路径变得简单。这篇研究论文提供了一组合成程序来确定精确的评估者的方法。文章还总结了在同行评审期刊和会议记录中发布研究论文的过程，旨在为早期阶段的研究人员提供一个进入评估员基本问题的入口。本文采取跨学科立场，通过实施几个算法（如 AFFA（作者频率查找算法）、ACFA（作者引用查找算法）、ARA（作者排名算法）和 AEFA（作者电子邮件查找算法））从增强技术知识研究中给出了丰富的例子。

II. 相关工作

同行评审是由合格专家对研究的竞争力、重要性和原创性进行评估的过程 [1]。许多期刊有一个编辑团队，包括主编和几位科学编辑，他们被分配负责个别论文的同行评审任务 [2]。这些期刊有时会在选择审稿人前提供讨论环节。它们只是寻找主题专家来挑选审稿人 [3]。但是他们的专家名单非常有限。有时候找到一位主题专家是非常困难且耗时的。一般来说，文章分发给特定评估者的技巧可以归类为两个模块：基于偏好的方法和基于主题的方法。

A. 基于偏好的方法

评估员的选择方法众多，使得难以使用评估员的偏好或受压数据。基于记录偏好的方法要求评估员提交论文以了解他们是否对这些论文感兴趣。这种方法的软肋在于命令统计的低效性。Rigaux [5] 建议使用协同过滤技术通过让使用者在给定主题内的大多数论文上出价

来提高偏好。集体强化程序的基本假设是，对于匹配文章提出类似意见的评估员可能对额外的论文有相似的偏好。

B. 主题导向方法

对论文分配给评估者的职责的一种解释是，文章必须分配给具有该主题一定程度知识的评估者。这种理解导致了利用有利信息的主题基础实践。通过使用这些数据，评论员的任务可以被安排以确认论文中心与分析员研究领域的相似程度。根据主题知识对每个评论员的重要排名，基于一篇特定论文的价值观，被称为专家发现或能力塑造。这种方法中一个被激发的问题是识别论文所涵盖的问题。Dumais 和 Nielsen[4] 通过在审稿人提供的摘要上训练的潜在语义索引将论文匹配给审稿人。在 Basu 等人 [5] 的研究中，从可能的审稿人撰写的论文中提取了摘要并通过搜索引擎从网络获取，然后构建了一个向量空间模型进行匹配。Yarowsky 和 Florian[6] 通过一个类似的向量空间模型结合朴素贝叶斯分类器扩展了这一想法。Wei 和 Croft[7] 提出了使用带有狄利克雷平滑的语言模型的主题基础方法。

III. 方法论

相关工作讨论了由编辑委员会手动进行的审稿人选择。但是，我们的目标是通过实现包含 AFFA、ACFA 和 AEFA 等算法的软件或程序来自动完成这项工作，以检索到正确的人员来评审研究论文。

A. 投影 AFFA

我们需要从论文的参考文献部分提取作者名称和频率。但，PDF 文件无法被编译器读取，而文本文件则可以轻松被编译器读取。我们正在寻找基于 Java 的 API 将 PDF 转换为文本，并在查阅多个博客后；PDFBox 项目进入了我们的视野。PDFBox 是一个借阅库（注：原文*lending library*可能需根据上下文调整翻译），能够处理多种类型的 PDF 文档，包括已转换的 PDF 格式并提取文本，同时还提供命令行工具将 PDF 转换为文本文件 [8]。

1) 逻辑分割参考部分以获取作者姓名： 由于我们想要从研究论文的参考文献部分找到作者，并且观察到每个参考文献部分都通过关键词 “REFERENCES”（大写）或 “References”（首字母大写）分隔。因此，我们尝试通过调用函数 Split（“REFERENCES”）来分割参考文献部分，如果未找到两部分，则再试用 Split（“References”）。

2) 应用正则表达式查找作者姓名： 正则表达式帮助我们将参考文献拆分，并提供如图 1 所示的格式。我们观察到每个参考文献包含作者姓名、研究标题、期刊名称、卷号、页码、年份以及一些换行和不必要的值。

[1] J. Mo and R. Heath, "High SNR capacity of millimeter wave MIMO systems with one-bit quantization," in Proc. IEEE Information Theory and Applications Workshop (ITA), 2014, pp. 1-5.
Proc. IEEE International Conference on Communications (ICC), 2015.

图 1. 参考文献书写格式（引用）

由于我们的目标是从论文中找出作者，这些作者信息不会包含任何回车、换行符和数字。因此，必须以相同的格式提取所有引用以便机器理解。因此，我们应用了一些其他的正则表达式来移除回车、换行符和所有的数字，并将引用获取为相同结构。现在，所有的引用都具有相同的格式，这是我们期望的格式。如果我们查看每个引用的结构（图 1），会发现每个作者之间用逗号 (,) 分隔，论文标题以双引号 (“) 开头。然后我们调用了一个新方法来对接哈希表，其中字符串类型用于存储作者的名字，整型用于频率。HashMap 类使用哈希表实现 Map 接口 [9]。这使得基本操作如 get() 和 put() 的执行时间即使对于大型集合也能保持恒定。此过程将每个引用中通过逗号 (,) 分隔的所有部分映射到一起。然后，我们将每个引用中的包含逗号的部分拆分并存储到哈希表中。之后，我们深入观察发现很多部分被逗号分隔开。因此我们的挑战是只提取出作者的名字。为此，我们引入了一个名为 checkStringValidity()[9] 的方法来找出有效的作者名。如果名字以双引号（“）开头，则我们将它从哈希表中删除。因为，我们知道论文标题是以双引号（“）开头的，并且如果我们发现名称包含字典中的非姓名单词时也会将其从列表中删除。

3) 统计作者频率： 获取有效作者名单后，我们就可以开始通过哈希表进行作者频率统计。在哈希表中找到新名字时，则将值设置为 1。如果我们在哈希表中找到了一个已有的作者，并且它包含整数值，我们将该值加一。此过程适用于哈希表中的所有作者。分配完每个作者的频率值后，调用函数 entriesSortedByValues() 按照频率对作者进行排序 [10]。最后我们得到了如图 2 所

示格式的结果。

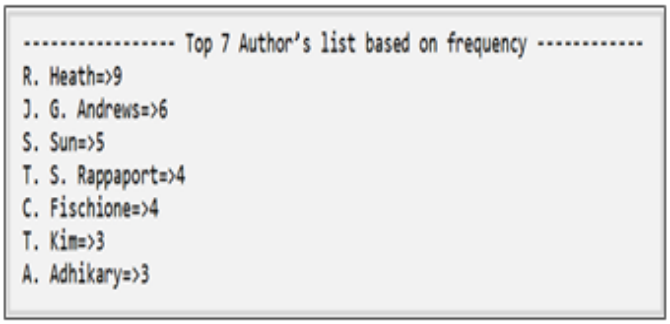

图 2. 作者列表及其频率分布

4) 投影 ACFA: 现在，我们的目标是找出选定作者的出版物权重。为此我们尝试从 Microsoft Academics、Research Gate 和 Google Scholar 账户获取信息。尽管经过长时间观察，我们只能访问 Google Scholar 账户，而另外两个的数据则绑定在了 web 服务器上。通过仔细观察发现，仅能通过以下 URL 访问 Google Scholar 账户，其中作者名字的部分由加号 (+) 运算符连接如图 3 所示。如果我们仔细观察图 3，可以看到只有作者名字

https://scholar.google.com/scholar?hl=en&q=R.+Heath&btnG=&as_sdt=1%2C5

https://scholar.google.com/scholar?q=S.+Sum&btnG=&hl=en&as_sdt=0%2C5

https://scholar.google.com/scholar?q=C.+Fischione&btnG=&hl=en&as_sdt=0%2C5

图 3. 查找选定作者的学者账户的搜索格式

那部分被加号 (+) 运算符替换。因此，我们可以通过简单地将链接中那一部分替换成所需作者的名字来轻松搜索到该作者。经过对作者 Google Scholar 账户源代码的长时间审查，发现每位研究者都有一个由 12 个字符组成的唯一 ID，这个 ID 由大写字母、小写字母、下划线和数字组成。现在我们的挑战是找出每个研究者或账户持有者的唯一 ID。通过使用正则表达式进行逻辑分割后，在页面源代码中发现了该编号如表 1 所示。

获得唯一 ID 后，我们使用了一个名为 getPublicationInfo(Id) 的方法。该方法会重定向到作者的 Google Scholar 账户。在那里，我们找到了学者姓名、大学信息、引用次数、h 指数和 i10 指数。然后，我们的挑战是检索与作者相关的所有信息 [13]。为此，我们将源代码拆分成了几个逻辑点，并使用了正则表达式。最后，我们在表 2 所示的格式中得到了所有结果。

表 I
作者姓名及 Google 学术账号 ID

Author's Name	ID (Google Scholar)
R. Heat	W_ZpqUwAAAAJ
S. Sum	rrfl7UsAAAAJ
C. Fischione	RWGj7esAAAAJ

表 II
作者列表及从网页提取的出版信息

Name (Short)	Name	University	h-index	i10-index	Citations
R. Heat	James Robert Heath	California Institute of Technology	99	220	54428
S. Sum	Shao-Cong Sun	Unknown Affiliation	63	128	13624
C. Fischione	Carlo Fischione	Associate Professor, KTH Royal Institute of Technology	23	48	1923

5) 投影 AEFA: 现在，我们有了作者及其发表权重的完整列表，但主要挑战是从网络中提取他们的电子邮件。一些挑战包括，除非作者在 Google Scholar 账户中链接了其个人资料，否则很难找到主页或学术档案。有时，在互联网的数据海洋中也难以找到正确的人。有些作者绑定电子邮件地址，而有些则将其保存为图片，这几乎是不可能被编译器读取的。我们知道，一个电子邮件地址可能包含大写字母（A-Z）、小写字母（a-z）、数

字（0-9），以及一些特殊字符如点 (.)、下划线 (_)、短横杠 (-)，并在中间包含 @ 符号。我们只是试图运用这个概念来解决所有挑战并克服其中的一些问题。最终，通过应用正则表达式（RE）代码能够找到一些作者的电子邮件地址。

IV. 实验结果分析

为了实验，我们选取了三篇已发表的研究论文，并将结果（前三作者）分别显示在表 3、4 和 5 中，对应数据集 1、2 和 3。

1)： A. Akusok, K. M. Bjork, Y. Miche 和 A. Lendasse, “高性能极限学习机：大数据应用的完整工具箱,” IEEE 交易, 发布于 2015 年 6 月 30 日, 2015 年第 3 卷

2)： B. Kehoe, S. Patil, P. Abbeel 和 K. Goldberg, “云机器人与自动化综述,” IEEE 自动化科学与工程汇刊, 第 12 卷, 第 2 期, 2015 年 4 月

3)： K. Liu, L.Xu 和 J.Zhao, “基于词对齐模型的在线评论意见目标和意见词汇共抽取” IEEE 知识与数据工程汇刊, 第 27 卷, 第 3 期, 2015 年 3 月

表 III
数据集 1 的最终输出

Author's (Rank by Score)	Total Score	Verified Email Domain	Homepage Link	Email
G. B. Huang	1649	ntu.edu.sg	http://www.extreme-learning-machines.org	
A. Lendasse	408	uiowa.edu	http://www.engineering.uiowa.edu/mie/faculty-staff/amaury-lendasse	lendasse@uiowa.edu
M. van Heeswijk	44	aalto.fi	http://users.&.tkk.fi/heeswijk48	

V. 性能分析

条形图 1 展示了我们从程序中获得的三个实验数据的整体结果。

表 IV
数据集 2 的最终输出

Author's (Rank by Score)	Total Score	Verified Email Domain	Homepage Link	Email
K. Goldberg	1116	berkley.edu	http://goldberg.berkeley.edu/	goldberg@berkeley.edu
M. Beetz	845	in.tum.de	http://ias.cs.tum.edu/people/beetz	lendasse@uiowa.edu
R. D'Andrea	637	ethz.ch	http://www.raffaello.name/	

表 V
数据集 3 的最终输出

Author's (Rank by Score)	Total Score	Verified Email Domain	Homepage Link	Email
H. Wang	7682	nshs.edu	http://www.feinsteininstitute.org/Feinstein/About+Haichao+Wang	xwang8@sjtu.edu.cn
K. Liu	4545	northwestern.edu	http://www.tam.northwestern.edu/wkl/	
Z. Liu	3936	tenorth.de	http://www.tenorth.de/	

我们从条形图 1[图 4] 和线图 [图 5] 中看到，频率查找的准确率为 100%，ID 为 95%，电子邮件域为 86%，专业信息为 95%，H 指数为 90%，i10 指数为 90%，引用次数为 90%，主页为 71%，电子邮件为 29%，错误 ID 选择仅为 10%。整体准确率超过 80%。因此，我们可以轻松地说，这是一个很好的自动寻找审稿人的工作。

最后，我们可以从条形图 2 中看到频率查找的准确

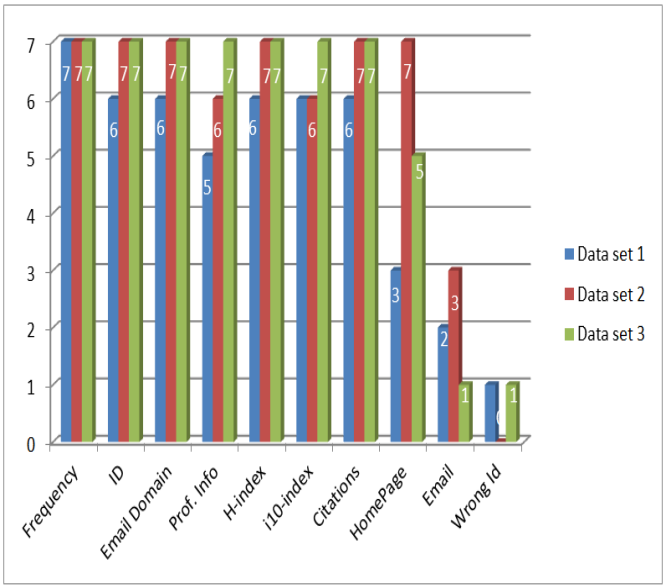

图 4. 条形图 1：数据集 1、2 和 3 的输出分析

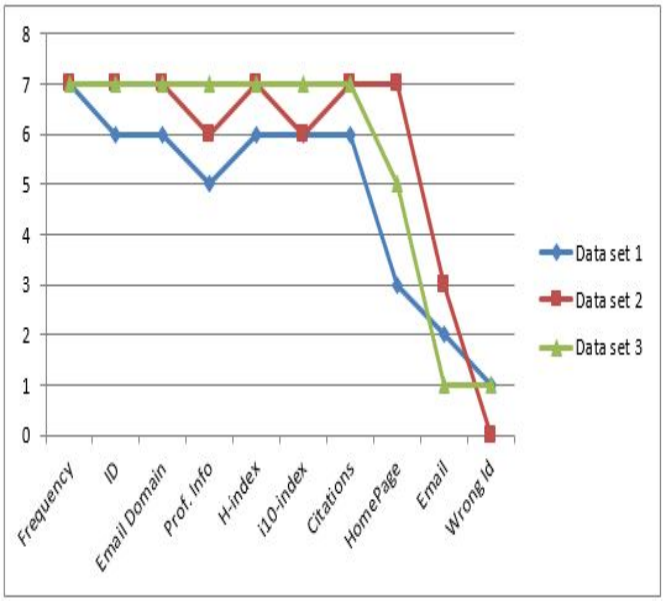

图 5. 折线图 1：数据集 1、2 和 3 的输出分析

性是 100%，ID 是 95%，电子邮件域名是 86%，专业信息是 95%，H 指数是 90%，i10 索引是 90%，引用次数是 90%，主页是 71%和电子邮件是 29%。而错误的选择 ID 仅占 10%，整体准确性超过 80%。因此，我们可以很容易地说，这是一项自动查找审稿人的出色工作。性能分析还告诉我们每个属性的正确值数量。

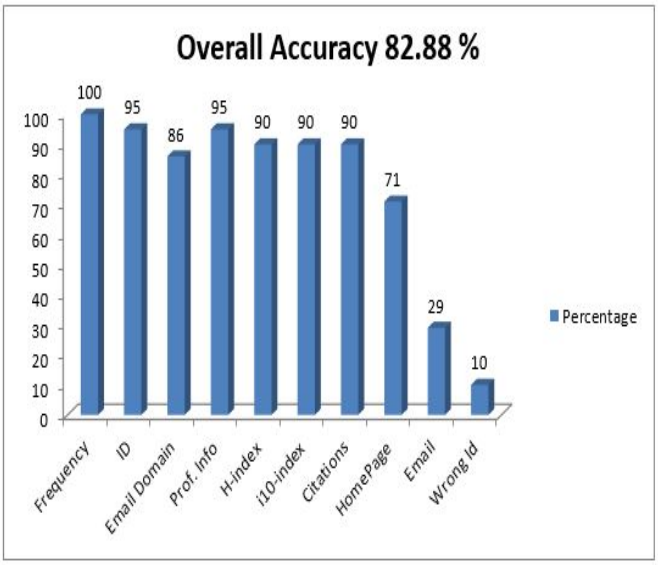

图 6. 条形图 2：数据集 1、2 和 3 最终输出的准确性

VI. 结论

这篇论文的工作对于像 IEEE、ACM、Springer 和 IGCA 等期刊非常有用，可以帮助自动找到评审员来审阅研究论文。确实，对主编来说，手动为研究论文寻找最佳评审员是一项非常艰巨的任务。为了找到研究论文的评审员，主编可能需要遵循一个冗长的手动过程。

- 选择了论文关键词以了解研究领域的方向。
- 手动搜索他们的审稿人列表以找到该领域的教授。
- 如果在审稿人名单中未找到。则在线搜索同一领域的教授。
- 最后，选择审稿人。该过程耗时。

我们只是想通过应用一些算法来自动化这个过程，这样任何研究期刊的主编都可以使用此程序或软件在几分钟内自动找到带有电子邮件的最佳评审员。甚至个别研究人员也可以在发表之前审阅自己的论文。已经完美地完成了一个具体的概念验证实现，用于找到最佳的研究论文评审员，并通过性能分析图表进行了验证。

参考文献

[1] B.T.Sense. (2004) Peer review and the acceptance of new scientific ideas. [Online]. Available: http://archive.senseaboutscience.org/data/files/resources/17/peerReview.pdf

[2] I. Hames, *Peer review and manuscript management in scientific journals: guidelines for good practice.* John Wiley & Sons, 2008.

[3] R. Cooley, B. Mobasher, and J. Srivastava, "Web mining: Information and pattern discovery on the world wide web," in *Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on.* IEEE, 1997, pp. 558–567.

[4] S. T. Dumais and J. Nielsen, "Automating the assignment of submitted manuscripts to reviewers," in *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1992, pp. 233–244.

[5] C. Basu, H. Hirsh, W. W. Cohen, and C. Nevill-Manning, "Technical paper recommendation: A study in combining multiple information sources," *Journal of Artificial Intelligence Research*, vol. 14, pp. 231–252, 2001.

[6] D. Yarowsky and R. Florian, "Taking the load off the conference chairs-towards a digital paper-routing assistant," in *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

[7] X. Wei and W. B. Croft, "Lda-based document models for ad-hoc retrieval," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006, pp. 178–185.

[8] P. T. Parser:. (2004) Converting pdf to text in java using pdfbox. [Online]. Available: http://www.prasannatech.net/2009/01/convert-pdf-text-parser-java-api-pdfbox.

[9] S. K. Dey and T. Al Mahmud, "Program complexity finder: A tool for finding program complexity in terms of cognitive weight based on complexity measurement algorithm," *International Journal of Computer Applications*, vol. 131, no. 8, 2015.

[10] S. Barua, T. Al Mahmud, S. K. Dey, and M. M. Rahman, "Future of social network with collaboration of cloud computing and brain computer interface," *Future*, vol. 145, no. 14, 2016.