

# Introduction

---

跨学科合作已成为当今学术研究中必不可少的组成部分。在全球化和信息技术的发展推动下，越来越多的学者需要与其他领域的研究者合作，从而产生更具创新性和综合性的研究成果。

不同领域的融合代表了科学进步和创新的一个有希望的领域，在这里最有可能发生突破和变革。来自不同学科的学者之间的合作努力对于促进科学发展至关重要。因此，需要一个系统的方法来识别和推荐潜在的跨学科合作者。这包括发现和评估现有的跨学科网络，并根据学者们的具体描述推荐合适的合作伙伴，以促进和推动有效的跨学科合作。

然而，跨学科合作也面临着一些挑战，如学科专业性、知识界限和沟通障碍等。传统的合作方法通常基于一个特定领域的知识、经验和社交网络，无法发现和利用那些来自不同学科领域但具有潜在合作关系的学者。因此，开发具有高精度和实时性的跨学科合作发现和推荐系统至关重要。

传统的发现和推荐学者之间的合作的方法主要是基于他们在特定领域内的研究方向和内容。这种方法通常依赖于分散在学者所属机构，学者主页和各个学者数据库中的学者研究领域，职称，学术成就，工作经历等数据。这些数据需要大量的人工标注和维护，并且难以保证数据的完整性和实时性。而且由于不同机构和学者数据库的数据格式不同，且不完全对外开放，集成这些数据也是一个很大的挑战。

另外还有方法依赖于多模态的数据，例如课件和课堂录音数据等，这种方法的成本过高且不具有普适性。因此，传统的学者合作发现和推荐不能促进对学者之间合作关系的深入探索，也不能解决跨学科合作的迅速发展所带来的挑战。

To tackle the aforementioned challenges, 本文介绍了一种名为 CollabXplorer 的系统，用于跨学科合作发现和推荐。我们知道，为了为学者推荐合适的跨领域学者合作对象，我们必须要先以一个合适的方式评估学者的研究方向。因此本文首先聚焦于学者的量化和表示工作，即将学者嵌入到更低维连续向量空间。具体而言，我们提出了一个创新性的基于全局语义信息和社会影响力的学者嵌入模型，用于得到 vectors for author representation。我们的学者嵌入模型的输入由全局语义信息和社会影响力两部分组成。其中，语义输入指学者全部论文的摘要，代表了学者在各个学科全部学术成果。语义输入的引入这使得我们可以跨越学科间的壁垒，从更精细的语义维度上，对来自多个学科的学者进行通用性的评估；社会影响力输入是学者在各个学科内影响力和贡献力综合度量，它反应了学者的专长以及在交叉学科的合作情况。同时，以上两个输入是根据论文摘要，论文发表量、论文被引频次等数据实时计算的，使得我们可以实时追踪学者的研究动态。

在成功地获取到学者向量之后，我们利用现有的数据发现潜在的学者间跨领域合作关系并推荐满足学者（用户）需求和偏好的合作者。为了实现这一目标，我们提出了一种算法，通过考虑学者间的 suitability，合作者的个人综合影响力以及用户输入的提示词，向用户推荐匹配的合作者。此外，CollabXplorer 还提供了用户界面，以帮助用户更好地与系统交互，发现目标研究领域的合作者。该系统采用了多层次架构和现代软件技术，确保系统的性能和稳定性。本文对系统的设计和实现进行了详细介绍，并邀请来自北京邮电大学 (BUPT) 的专家和学者对本系统进行了评估和体验。实验数据有力地验证了 CollabXplorer 的准确性和有效性。

总的来说，本文为学者与研究人员提供了一种有效的方法来发现和利用跨学科合作的机会。CollabXplorer 的方法和算法可为学术界提供更加实用和高效的工具，以促进更多的跨学科合作和创造性的研究成果。

## Related Work

---

一直以来，推荐系统一直是许多研究人员的热门关注，它的可用性体现在各个领域。同样的，推荐系统在教育领域中也发挥着重要的作用。

在跨学科领域合作问题上，我们看到过不同的解决方案。其中 XXX 的系统专注于推荐最适合的交叉专业 `\cite{}`，简化和纠正决策者的工作，促进交叉专业改革。但是这种情况下存在的现实问题是没有更多考虑这些领域下学者的具体合作需求，盲目的领域交叉可能会导致科研上的学者协作不足，浪费科研资源等问题。另一方面，该系统需要论文，课间和课堂录音数据多种模态的数据输入，同时需要大量的人工标注数据，不适合大规模应用于全球的机构和学科领域。

另外 XXX 的推荐系统专注于同类型学者的推荐 `\cite{}`，输入作者信息以及相关学者信息后，他们根据各种因素对学者进行排名，如论文内容、期刊质量等，并在此基础上生成评价分数，进行权重分配后计算学者相似度来进行学者匹配。但是对于跨领域合作的要求而言，它的不足之处在于我们获得的只有相似领域的学者推荐，但对于跨领域的潜在合作发现仍有很大不足。

CollabXplorer 的核心优势在于，我们从单词的粒度上全面地分析了学者的全部公开文献，并且在推荐合作者时考虑了用户的需求和偏好。以上两点创新赋予了 CollabXplorer 更强的普适性和可扩展性，使得它可以应用于广泛的学科领域，并且切实服务于学者的跨学科合作需求。相较于目前的学者推荐系统：

- 我们聚焦于学者所有文献的摘要，并且考虑学者的综合影响力、合作网络，构建了一个更加全面的学者向量表示方法。
- 我们不需要分散于各处的学者研究领域，职称，学术成就，工作经历等结构化数据。这些数据需要大量人工维护，难以集成，且无法保证数据的完整性和实时性。因此我们跳过传统的特征提取工作，直接根据论文内容将学者嵌入到低维的向量空间进行后续分析。我们选择了一些开放的论文数据库作为数据源，保证了数据的完整性和实时性。
- 我们不需要为特定的学科领域构建学者评估和推荐方法。通过我们的学者嵌入模型，得到的学者向量可直接用于合作者推荐，适用于广泛的学科领域。
- 我们不需要大量的人工标注数据，来训练推荐模型，只需要用户输入提示词，就可以为用户推荐符合描述且契合度高的（跨学科）合作者。

## CollabXplorer

本节将介绍 CollabXplorer 的系统设计、实现和交互设计。

系统设计及实现部分重点介绍了 CollabXplorer 的系统架构、数据来源、学者嵌入模型，学者推荐算法以及关键组件的实现；用户交互体验部分主要涵盖系统的界面设计、功能和用户体验。

系统主要目标是从大量非结构化的学术数据中发现有意义的跨学科合作潜力，并将其通过推荐算法以可视化的方式呈现给用户。

## System Design and Implementation

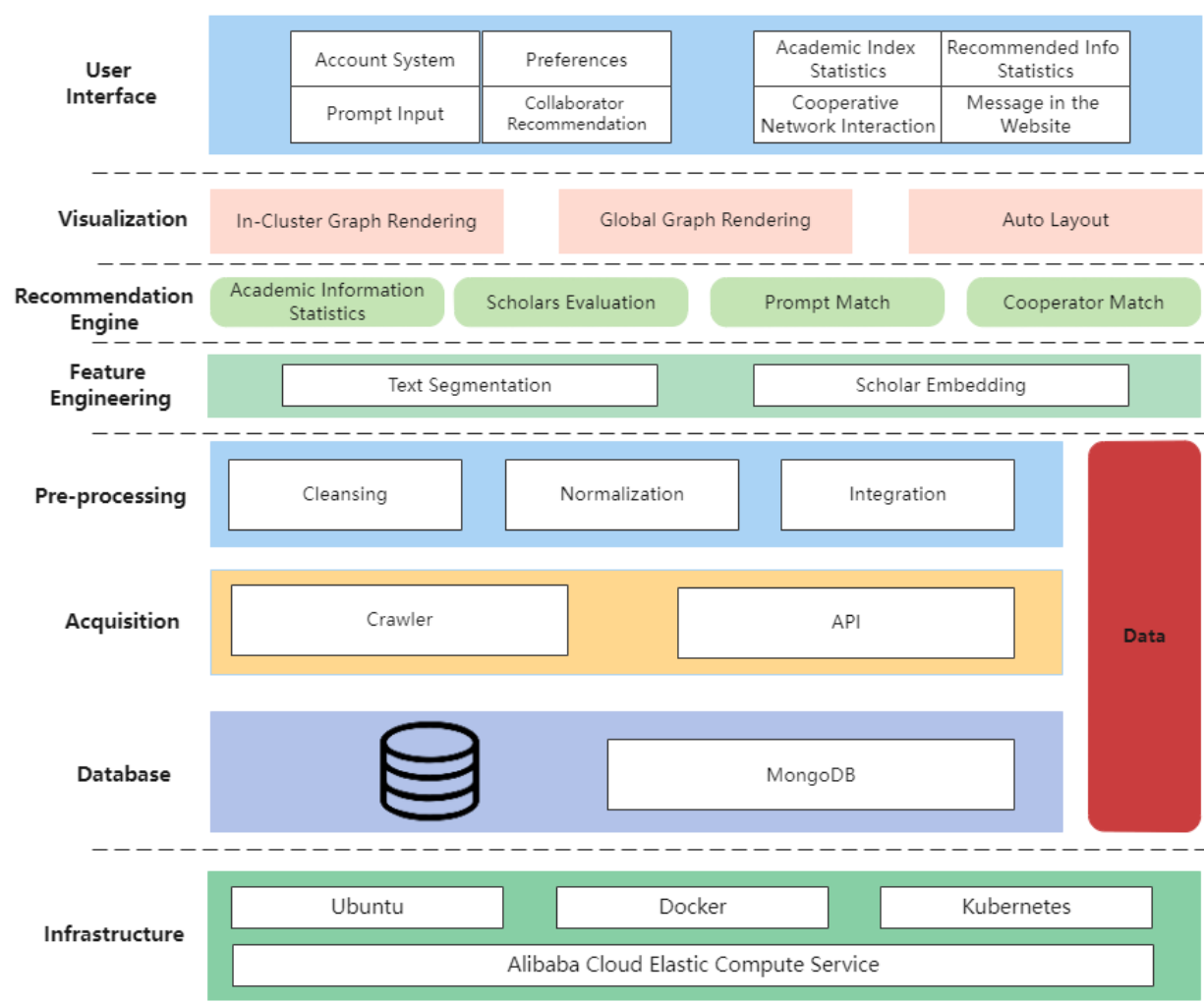
本系统的架构 `\ref{fig:framework}` 主要包括以下层次：基础设施层、数据层、算法层和应用层。数据层主要包括数据存储、获取和预处理；算法层涵盖针对本任务的特征工程，推荐引擎和合作网络的可视化。应用层包含账号系统，用户偏好，学术/推荐信息统计，提示词输入，合作者推荐，合作网络交互和站内互动等系统功能以及用户界面。

关于基础设施，我们在阿里云弹性计算服务的平台上，使用了 Docker 容器技术，将系统的各个组件打包成镜像，方便部署和管理。并且使用了 Kubernetes 集群管理工具，实现了系统的自动化部署，维护和扩展。

本系统采用前后端分离的开发模式，其中前端使用 Vue.js 框架开发，后端使用 Python 的 Flask 框架实现。一个典型的基于用户提示词的推荐任务的系统工作流程可以概括为：

- 前端：
  - 通过调用后端 API 实现用户登录、注册、身份认证等功能。

- 通过后端 API 将用户输入的提示词传递给后端。
  - 调用后端 API，将用户输入的提示词以参数形式传递给后端，阻塞进程等待后端返回结果。
- 后端：
  - 根据用户的提示词和推荐偏好，计算所有数据库内所有学者的向量。
  - 计算所有学者与当前用户的 suitability，并降序排序推荐排名靠前的学者。
  - 将推荐结果封装成 JSON 格式并返回给前端，并持久化本次记录。
- 前端：
  - 收到并解析 JSON 数据，刷新页面推荐统计信息和推荐列表。
  - 在本地渲染跨学科合作网络。
  - 监听用户的交互行为，如点击推荐列表中的学者，点击跨学科合作网络中的节点等，调用后端 API 将用户的行为记录持久化。



## Data Layer

学者的学术数据是设计本系统的时重要考虑因素，因为数据的质量是推荐算法的关键。`\ref{fig:data_layer}`介绍了数据层的设计和实现。

Data Layer	Pre-processing	Cleansing	Normalization	Integration
	Acquisition	Crawler	AAAI	ACM
			Google Scholar	
		API	IEEE	arXiv
			ScienceDirect	Springer
			dblp	Crossref
			ORCID	
		Database	MongoDB	

## Data Acquisition

我们使用 ORCID 作为学者的唯一标识。ORCID 是一个开放的、免费的、持续增长的学者标识系统，它为每位学者提供了一个唯一的、持久的、可验证的学者标识符。

然而，ORCID 并不直接提供论文的元数据，例如摘要，关键词等，因此我们需要通过其他渠道获取学者的学术信息。Crossref 提供了全球大部分的学术出版物的元数据。我们根据 Crossref 提供的 API，以学者的 ORCID 作为参数查询，可以获得该学者的全部论文的元数据，包括全部作者，标题，摘要，关键词等信息。对于未在 ORCID 上查询到的学者，我们的备用数据库是 dblp 和 Google Scholar。通过查询学者的名字，我们可以获得该学者每篇论文的 URL。我们编写了相应的网页爬虫或使用出版社提供的 API，根据 URL 获取论文的元数据。

我们选择更为灵活的非关系型数据库 MongoDB 作为数据存储，MongoDB 是一个基于分布式文件存储的数据库，它的数据模型是面向文档的，数据以 JSON 形式存储，因此非常适合存储学者的学术信息。

我们定义每位学者的文档：

```

1  {
2    "orcid": "xxxx-xxxx-xxxx-xxxx",
3    "dblp_id": "xxx/xxxx",
4    "name": ["name1", "name2", "..."],
5    "affiliation": [["dept.1", "Aff.1"], ["dept.2", "Aff.2"], "..."],
6    "articles": {
7      // 1st article's metadata
8      "xxxxxx1": { // DOI or dblp_id
9        "title": "article's title",
10       "url": "article's url",
11       "year": "year of publication",
12       "journal": "journal's name",
13       "authors": ["name1", "orcid/dblp_id"], ["name2", "id"], "..."],

```

```

14     "abstract": "xxxxxx",
15     "keywords": ["keyword1", "keyword2", "..."],
16     "citations": "n"
17 },
18 // 2nd article
19 "xxxxxx2": {"...": "..."},
20 "...": {"...": "..."}
21 }
22 }

```

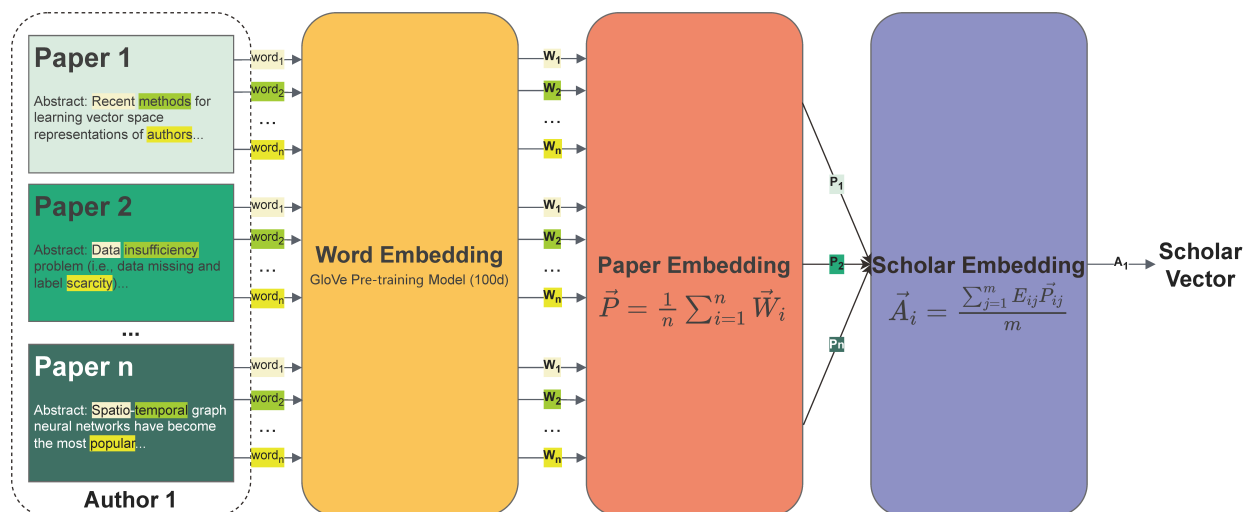
## Data Preprocessing

下一步，我们需要对学者的学术信息进行预处理，以保证数据的质量。我们的预处理主要包括以下几个步骤：

1. 数据清洗：去除异常数据，例如，某些论文缺失摘要，关键词，合著者等信息。
2. 数据规范化：去除停用词，停用词是指在自然语言处理中，由于其频繁出现而没有实际意义的词，例如，the, a, an, of, in, on, etc. 这些词在文本中出现的频率很高，但是对文本的含义没有太大的贡献，因此我们需要去除这些停用词。我们使用了 NLTK 库中的停用词列表，将数据规范化。
3. 数据集成：将来着不同数据源的学者的学术信息进行整合，包括冲突合并和去重。我们设置了定时任务，每天自动从 Crossref, dblp 和 Google Scholar 等数据源更新学者的学术信息，以保证数据的实时性。

## Embedding

为了提取学者的全部特征，然后进行学者推荐，需要将学者的学术信息转换为向量，我们称这一过程为学者嵌入。 \ref{fig:scholar\_embedding} 介绍了学者嵌入的工作流程。



词嵌入，也称为分布式表示，是自然语言处理(NLP)中广泛使用的技术，使机器能够捕捉单词的含义和上下文。Global Vectors for Word Representation(GloVe)就是这样一种方法，它因能够产生高质量的词嵌入而受到欢迎。

Pennington 等人 \cite 提出了 GloVe，一种基于语料库中单词的全局共现统计量来训练单词嵌入的方法。该方法包括使用奇异值分解(SVD)对单词共现计数矩阵进行分解，并学习捕捉单词之间统计关系的单词向量。

GloVe 中使用的关键公式是目标函数：

$$J = \sum_{i,j=1}^V f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log(X_{ij}))^2$$

其中 $V$ 是单词的词汇量， $X_{ij}$ 是单词 $i$ 和 $j$ 的共现次数， $w_i$ 和 $w_j$ 是词向量， $b_i$ 和 $b_j$ 是偏置词， $f(X_{ij})$ 是一个赋予罕见词对较小权重的权重函数。目标是通过调整词向量和偏差来最小化 $J$ 。词嵌入中另一个重要的公式是余弦相似度：

$$\text{similarity}(w_i, w_j) = \cos(\theta) = (w_i^T * w_j) / (||w_i|| * ||w_j||)$$

它通过计算两个词向量 $w_i$ 和 $w_j$ 之间夹角的余弦来度量它们之间的相似度。高余弦相似度值表明单词在语义上相似，并且在进行搜索的相似度计算时，余弦相似度具有优越性。<sup>\cite{cos2001item}</sup>

当把 GloVe 应用于本文章的问题时，即构建基于语义和综合影响力的合作网络，相应的原则是利用 GloVe 的预训练模型，将学者的论文摘要转化为向量，这个过程称为论文嵌入。然后根据各篇论文的综合影响力，对这些论文向量进行加权平均运算得到学者向量，这个过程称为学者嵌入。

## Paper Embedding

我们定义论文的向量 $\vec{P}$ :

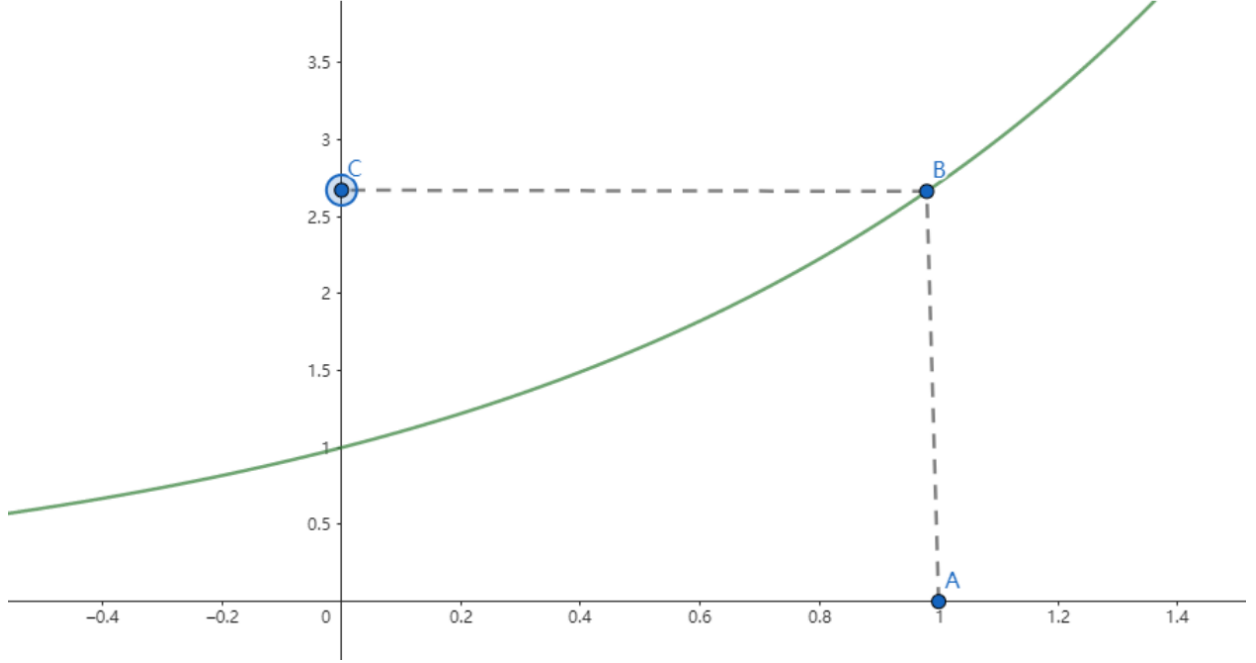
$$\vec{P} = \frac{1}{n} \sum_{i=1}^n \vec{W}_i$$

其中， $\vec{W}_i$ 是论文摘要中第 $i$ 个单词的词向量， $n$ 是论文摘要中单词的总数。这里我们使用了平均值来表示论文向量，这是因为我们认为论文的主题应该是由摘要中的所有单词共同决定的。

对于词向量 $\vec{W}$ ，我们使用 NLTK 库对数据库中的论文摘要进行文本分割，然后将得到的单词，也就是 token，映射到 GloVe 的预训练模型中，得到每个单词的 100 维的词向量。在本文中，我们选择了 GloVe 的预训练模型 (6B tokens, 400K vocab, uncased, 100d vectors)，该模型是在维基百科语料库上训练的。

## Scholar Embedding

首先我们定义一些符号。 $x_{ij}$ 代表学者 $i$ 的论文 $j$ 中的合著者总人数。一般而言，对于一篇论文，作者的排名越前，他对于该论文的贡献度越大。因此，我们可以假设学者的贡献度与他在该论文中的排名成非均匀的正比关系，使用指数函数来表示（见图？）。



基于上述假设，则学者*i*对于论文*j*的贡献度 $E_{Rij}$ 表示如下：

$$E_{Rij} = \begin{cases} e, & k = 1 \\ e^{\frac{x_{ij}-k}{x_{ij}}}, & \text{others} \end{cases}$$

其中， $k$ 表示学者在论文中所有合著者中的排名。

此外，该论文的被引用次数能够反映该论文的影响力，通常情况下，被引用次数越多，该论文的影响力越大。因此，我们定义了学者*i*的论文*j*的影响因子 $C_{ij}$ ：

$$C_{ij} = \begin{cases} 0, & n_{ij} = 0 \\ 1, & n_{ij} > 0 \end{cases}$$

其中 $n_{ij}$ 是论文被引用的次数。并且结合论文的影响因子 $C_{ij}$ ，我们可以得到论文*j*的影响力 $E_{Cij}$ ，其定义如下：

$$E_{Cij} = C_{ij} e^{\frac{n_{ij}}{n_{i_{max}}}}$$

其中， $n_{ij}$ 是论文*j*被引用的次数， $n_{i_{max}}$ 是学者*i*的最高单篇被引用次数。

上述两个步骤分别提出了学者对于论文的贡献度和论文的影响力。接下来，我们基于 PageRank 思想上改进上式，PageRank 是由 Sergey Brin 和 Lawrence Page 等人 [cite?] 提出的在考虑网络单点的影响力基础上，考虑其他因素的影响力的算法，我们将模型进行了调整：

$$E_{ij} = \lambda E_{Rij} + (1 - \lambda) E_{Cij}$$

$E_{ij}$ 代表论文*j*的综合影响力，这里我们考虑两个因素：学者*i*对于论文*j*的贡献度 $E_{Rij}$ 和论文*j*的影响力 $E_{Cij}$ ，其中权重参数 $\lambda$ 决定了 $E_{Rij}$ 和 $E_{Cij}$ 两个因素对 $E_{ij}$ 的影响情况，取值范围为 $[0, 1]$ 。

通过上述步骤我们得到了学者*i*在论文*j*中的影响力 $E_{ij}$ ，接下来将其与 GloVe 提取的学者*i*的论文*j*的向量 $\vec{P}_{ij}$ 相乘，并进行加权平均运算，得到学者*i*的向量 $\vec{A}_i$ ，其定义如下：

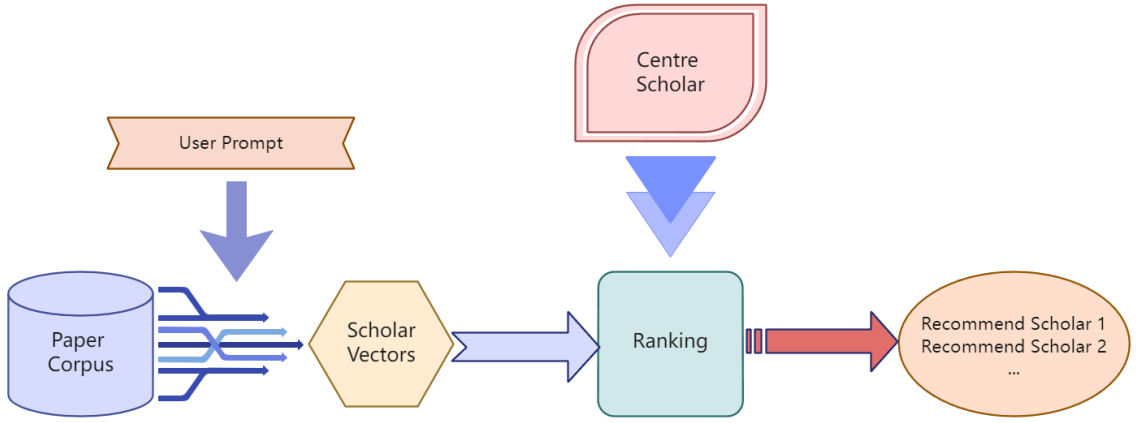


$$\vec{A}_i = \frac{\sum_{j=1}^m E_{ij} \vec{P}_{ij}}{m}$$

其中,  $m$ 表示学者 $i$ 撰写的论文总数。通过此方法我们可以得到每一个学者的向量表示, 我们称上述过程为 author embedding。

## Collaborator Recommendation

对于推荐系统, 我们必须考虑到学者们的需求。在本文中, 我们允许中心学者(系统将为中心学者推荐合作者)输入 prompt 来描述他期望寻求的合作者。因此, 我们的目标是找到符合 prompt, 且与中心学者具有较高 suitability 的合作者。简而言之, 我们使用 prompt 调制了数据库中所有学者的向量, 进而计算出中心学者与其他学者的新向量的 suitability, 最终推荐排名靠前的学者作为合作者。我们推荐引擎的工作流程如图 \ref{fig:flow\_chart} 所示:



与公式 \ref{cite\_formula} 类似地, 我们定义用户输入的 prompt 的向量形式  $\vec{p}$ :

$$\vec{p} = \sum_{k=1}^n \vec{W}_k$$

其中,  $\vec{W}_k$ 指 prompt 中的第  $k$  个单词的 GloVe 向量表示。

接下来, 我们将学者  $i$  的论文  $j$  的向量  $\vec{P}_{ij}$  与 prompt 的向量表示  $\vec{p}$  进行相似度计算, 得到论文  $j$  与 prompt 的 similarity  $s_{ij}$ , 具体的计算公式如下:

$$s_{ij} = \cos(\vec{p}, \vec{P}_{ij})$$

由于  $s_{ij} \in [-1, 1]$ ,  $E_{ij} \in [1, e]$ , to change the range of  $s_{ij}$  from  $[-1, 1]$  to  $[1, e]$ , we modify the formula in the following way:

$$s'_{ij} = \frac{e-1}{2}(s_{ij} + 1) + 1$$

where  $s'_{ij}$  is the transformed value of  $s_{ij}$ .

进而, 我们得到学者  $i$  的论文  $j$  的权重  $W_{ij}$ , 由 similarity 和综合影响力  $E_{ij}$  \ref{cite\_formula} 两部分组成。

$$W_{ij} = \beta s'_{ij} + (1 - \beta) E_{ij}$$



其中，我们引入可变系数  $\beta$  权衡  $s_{ij}$  和  $E_{ij}$  的重要性。 $\beta$  可根据中心学者的需求变化。具体而言，如果中心学者更看重与合作者的 suitability，则取较大的  $\beta$ ；如果合作者的影响力更加重要，则取较小的  $\beta$ 。

至此，我们得到由 prompt 调制后的学者  $A_i$  的向量表示：（替换一句）

$$\vec{A}_i = \frac{\sum_{j=1}^m W_{ij} \vec{P}_{ij}}{m}$$

其中， $m$  表示学者  $i$  撰写的论文总数。

在此基础上，我们计算中心学者与学者  $A_i$  的 suitability  $S_i$ ：

$$S_i = \cos(\vec{U}, \vec{A}_i)$$

其中， $\vec{U}$  指中心学者的向量表示。

我们对  $S_1 \sim S_{126}$  进行降序排序，推荐排名靠前的学者作为中心学者的潜在合作者。

## Interaction Design



用户认证自己的学者身份后，选择允许本系统收录其学术数据，即可使用本系统。登录成功后，用户会进入 dashboard \ref{fig:dashboard}，该界面分为侧边栏控制台，展示界面与消息通知界面三部分，这里我们将主要介绍展示界面。

下面将按照从左至右，从上至下的顺序依次说明。首先是个人信息分区，在这里将展示用户的个人信息，如姓名，所属机构，邮箱等，下方可以为用户自行添加的学科及具体细分领域标签，上述信息均可修改，保存和撤销。

上方的四个矩形框分别是个人发表论文篇数、个人论文被引用次数、曾合作过的学者数量以及个人被系统推荐给其他学者的次数，这些指标将帮助用户以量化的形式评估自身的学术研究水平和合作水平。

在下面的四大分区中，首先是左上方的折线图区域，在当前图中，它指的是每月其他学者进行查询时你被推荐给他们的次数。左下方的柱状图区域指的是根据你的 prompt 推荐给你的学者，他们按照潜在合作可能性从上至下排列。

通过右上方的搜索框，用户可以输入非结构化的提示词，对其期望寻找的合作者的研究内容进行描述。系统会根据提示词，推荐与用户 suitability 较高的合作者。值得注意的是，合作者的学术影响力也是推荐指标之一，用户可前往 "Setting" 选项，调整个人偏好，即 suitability 优先或影响力优先，对应参数  $\beta$  的大小 cite 公示；另一方面，系统会对所有推荐的合作者进行分析，根据研究领域对他们进行分类和着色。相关统计信息呈现在 "Areas of Recommend Cooperators" 区域。

右下方的关系网络图是我们推荐系统的关键，在这张图中您可以可视化地观察您与推荐人员的潜在合作潜力，其中点代表的是学者，其大小随着合作潜力的变化而相应改变；连线的含义指两个学者之前作为合著者出现在同一篇论文中；点的颜色表示的是在当前搜索领域下的不同细分子领域，从而帮助使用者更好地选择。

总之，这个系统展示了用户的个人信息、学术研究水平以及推荐系统的功能。通过各种图表和可视化，用户可以深入了解自己的研究领域、合作学者以及潜在的合作可能性，从而更好地选择合作伙伴和方向。这个系统为学术研究者提供了一个交流和合作的平台，有助于推动学术研究的进步和发展。

## Evaluation

为了评估我们的系统，我们为来自 BUPT 的 126 位学者生成了学者文档 cite 代码 并写入数据库。这项工作得到了 BUPT 人事处和图书馆的支持，所有数据都经过了人工校对，确保了数据的准确性。针对本系统，我们设计了两个实验，分别是用来评估 author embedding 和推荐的效果。

### Embedding Evaluation

为了评估 author embedding 的效果，we conduct experiments on the author similarity task. Pennington [cite?] 等人对 GloVe 进行了 word similarity 评估，相应的，我们参考他们的工作设计了 author similarity task，这是因为我们的 embedding 是基于 GloVe 预训练模型的。具体而言，我们邀请北京邮电大学的专家，对我们数据库中 126 位学者的合作关系进行了评估。合作关系分为 5 个等级，分别为 1, 2, 3, 4, 5，对应 author similarity 的值为 0.2, 0.4, 0.6, 0.8, 1。其中等级 1 表示完全不合作，等级 5 表示密切合作。我们将这些评估结果记作  $t_{ij}$ ，其中  $i$  和  $j$  分别表示两个学者。

通过 author embedding，我们可以得到学者的向量表示。我们定义学者之间的相似度  $s_{ij}$ ：

$$s_{ij} = \frac{\vec{A}_i \cdot \vec{A}_j}{|\vec{A}_i| \cdot |\vec{A}_j|}$$

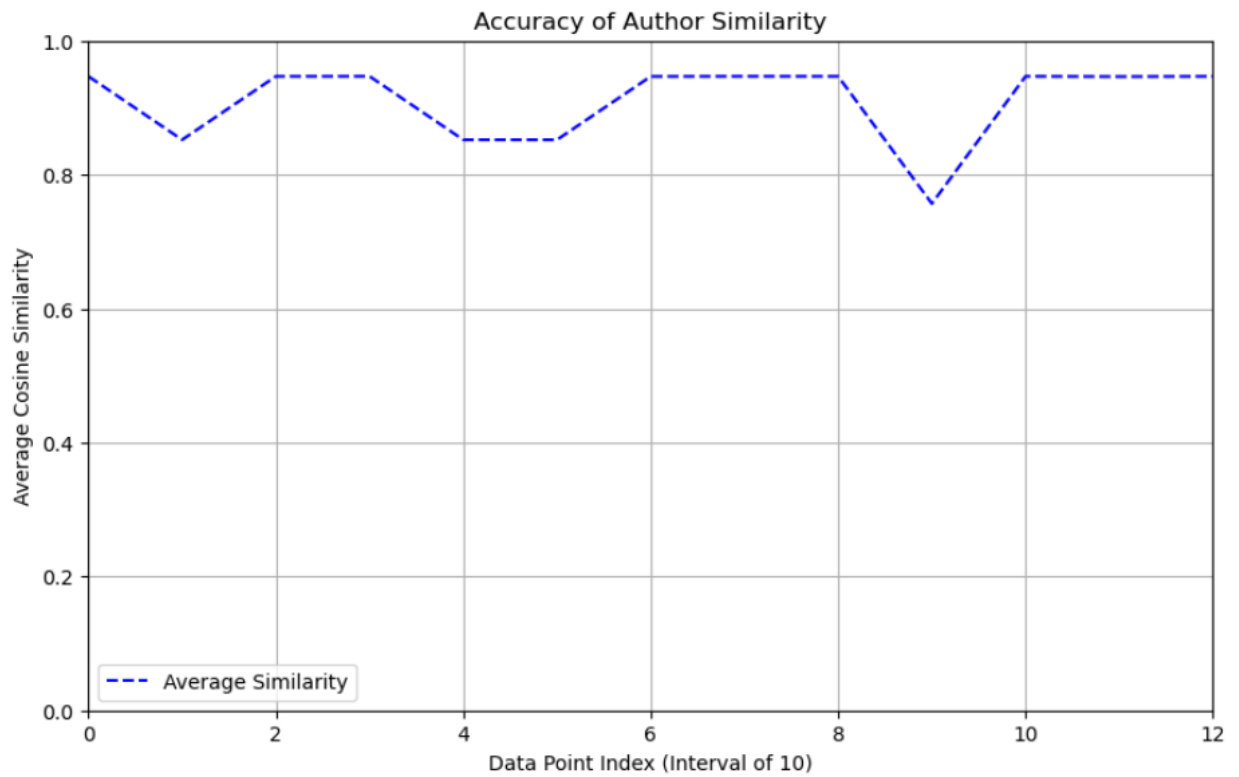
其中， $\vec{A}_i$  表示学者  $i$  的向量， $\vec{A}_j$  表示学者  $j$  的向量。

根据上述方法，同理可得学者  $i$  和其他所有学者的相似度  $\vec{s}_i$  和专家评估结果  $\vec{t}_i$ 。我们选择  $\vec{s}_i$  与  $\vec{t}_i$  的余弦相似度作为 author similarity task 的评估指标，越接近 1 表明我们的 author embedding 越准确。我们将这个指标记作 accuracy：

$$accuracy = \frac{\vec{s}_i \cdot \vec{t}_i}{|\vec{s}_i| \cdot |\vec{t}_i|}$$

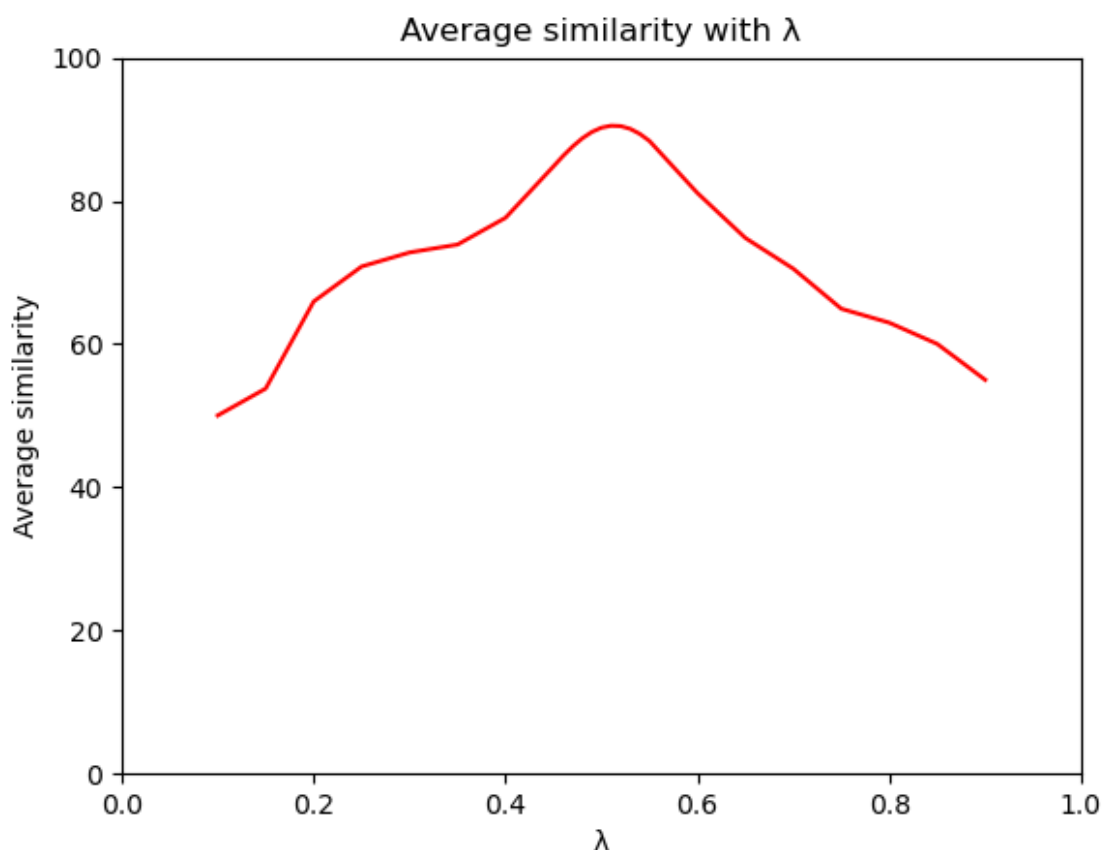
## 实验结果

我们在 Fig. 中展示了两组数据的相似度情况。我们选择以 10 个学者为步长，计算每 10 个数据点的 accuracy 的平均值。可以看到，accuracy 的平均值基本控制在 (0.75, 0.95)。通过计算，126 个数据点的 accuracy 的平均值为**0.91**，这表明我们的学者嵌入模型可以准确地反映学者间的合作关系。



## Model Analysis: $\lambda$ 的选择

In Fig 2, 我们展示了 Eqn. (?) 中，不同  $\lambda$  的选择对于整体模型准确度的影响。可以看出，开始时准确度随着  $\lambda$  的增加而上升。因为当  $\lambda$  的值越大时，作者的向量受其本身贡献度的影响增大，更加能反应出作者在该论文中的重要性，使得评价更加准确。但是在  $\lambda$  的值达到一定程度后，准确度就会开始下降。这是因为当  $\lambda$  的值越大时，作者的向量受论文影响力的影响增大，但是这个影响力是由论文的引用次数决定的，而论文的引用次数受到很多因素的影响，比如论文的发表时间，论文的主题等等，这些因素都会影响论文的引用次数。因此，当  $\lambda$  的值达到一定程度后，作者的向量就会受到很多不相关的因素的影响，使得评价不准确。因此，我们选择  $\lambda$  的值为 0.56，这个值能够使得模型的准确度达到最大值。



## Recommendation Evaluation

为了评估该推荐系统的效果，我们采取人工评分的方法，邀请了 20 名来自 BUPT 的研究人员对推荐结果进行评分。

### 实验设计

我们允许 20 位评分者自由选择 prompt，并指定中心学者。这使得评估者可以在自己熟悉的领域，根据专业知识和经验，评估推荐系统的准确性。

本系统根据评估者的输入的 prompt 和中心学者，推荐排名在前 5 的合作者，也就是生成一条推荐记录。评估者分别对这 5 位合作者进行满意度评分, where score  $s_j \in [0.00, 1.00]$ .

为了保证样本的多样性，我们要求每位评估者生成 5 条不同的推荐记录并进行评分，这意味着，每位评估者要评估 25 位合作者。因此，我们共收到了 100 条推荐记录的评分结果，包含对系统推荐的 500 位合作者的满意度评分。

本次评估中，系统的推荐偏好为 suitability 优先。表 \cite 展示了前 5 条推荐记录的评分结果。

	Recommendation Information				Score		
Record ID	Center Scholar	Prompt	1st	2nd	3rd	4th	5th
01	**	bioscience	0.54	0.91	0.39	0.89	0.36
02	**	graph neural network	0.70	0.62	0.49	0.37	0.52

	Recommendation Information				Score		
03	**	beamforming	0.85	0.84	0.51	0.69	0.26
04	**	multi-agent	0.82	0.76	0.74	0.71	0.40
05	**	econometrics	0.95	0.71	0.92	0.96	0.30

得到了每一个评分者对每一个推荐结果的评分后，我们定义如下 3 个评估指标：

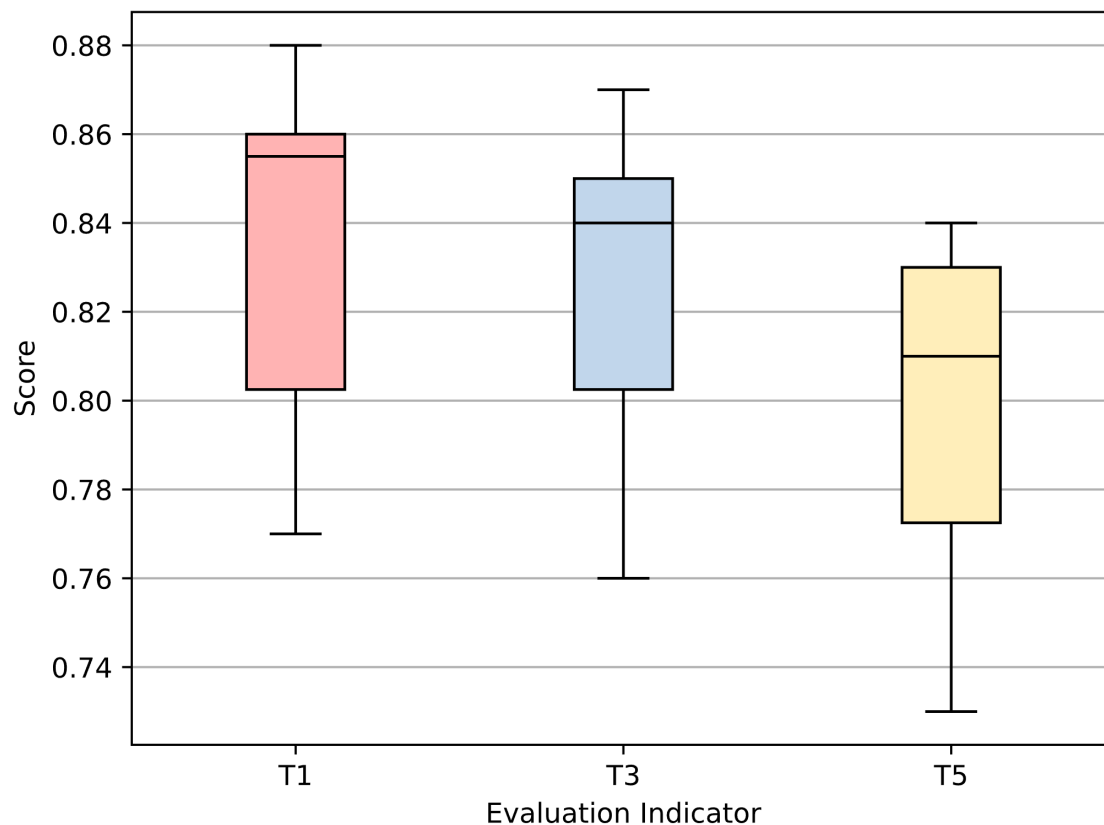
$$T_1 = s_1$$

$$T_3 = 0.7s_1 + 0.2s_2 + 0.1s_3$$

$$T_5 = 0.5s_1 + 0.3s_2 + 0.1s_3 + 0.05s_4 + 0.05s_5$$

其中 $T_i$ 代表前 $i$ 个推荐的评分， $s_j$ 代表评估者对于第 $j$ 个推荐的合作者的满意度评分。

我们采用箱线图对 100 个数据点（推荐记录）进行统计性描述，以便更直观地展示推荐系统的性能。箱线图的横轴为 3 个评估指标，纵轴为评分，箱线图的上下边界分别为 75% 和 25% 分位数，中间的线为中位数，箱线图的上下边缘分别为最大值和最小值。如下图 \ref{fig:boxplot} 所示：



$T_1$  的中位数和均值分别为 0.85 和 0.84，高于 $T_3$ 和 $T_5$ ，说明评估对系统推荐的第 1 位合作者的满意度普遍较高。这表明，在 suitability 优先的推荐偏好下，系统推荐的第 1 位合作者的与中心学者的契合度得满足了开展合作的需求；

$T_3$ 的中位数和均值与 $T_1$ 的差距在1%左右，可认为推荐的质量相近。另一方面， $T_3$ 拥有最小的方差，表明系统推荐的前 3 为合作者与中心学者契合度的相对稳定；

$T_5$ 拥有最低的中位数、均值以及最大的方差，表明推荐的质量和稳定性都在下降。这也是符合预期的，因为在推荐的前 5 位合作者中，合作者与中心学者的契合度在逐渐下降，部分合作者可能会影响到 $T_5$ 整体的推荐

质量。然而，为了保证推荐的多样性，我们在未来的工作中会尽量减少诸如此类的质量和稳定性下降问题。总的来说， $T_1 - T_5$ 的中位数和均值均都在 0.8 以上，这个结果表明，本研究中设计的推荐系统在推荐相关契合度高的合作学者方面具有较好的准确性，能够为学者推荐有价值的合作者。

## Conclusion

---

## Reference

---