algorithms

MDPI

Article

Information Retrieval and Machine Learning Methods for Academic Expert Finding

Luis M. de Campos [1,*], Juan M. Fernández-Luna [1], Juan F. Huete [1], Francisco J. Ribadas-Pena [2] and Néstor Bolaños [1]

[1] Departamento de Ciencias de la Computación e Inteligencia Artificial, ETSI Informática y de Telecomunicación, CITIC-UGR, Universidad de Granada, 18071 Granada, Spain; jmfluna@decsai.ugr.es (J.M.F.-L.); jhg@decsai.ugr.es (J.F.H.); nestor.bolanos@ugr.es (N.B.)

[2] Departamento de Informática, E.S. Enxeñaría Informática, Edificio Politécnico, Universidade de Vigo, 32004 Ourense, Spain; ribadas@uvigo.gal

* Correspondence: lci@decsai.ugr.es

Abstract: In the context of academic expert finding, this paper investigates and compares the performance of information retrieval (IR) and machine learning (ML) methods, including deep learning, to approach the problem of identifying academic figures who are experts in different domains when a potential user requests their expertise. IR-based methods construct multifaceted textual profiles for each expert by clustering information from their scientific publications. Several methods fully tailored for this problem are presented in this paper. In contrast, ML-based methods treat expert finding as a classification task, training automatic text classifiers using publications authored by experts. By comparing these approaches, we contribute to a deeper understanding of academic-expert-finding techniques and their applicability in knowledge discovery. These methods are tested with two large datasets from the biomedical field: PMSC-UGR and CORD-19. The results show how IR techniques were, in general, more robust with both datasets and more suitable than the ML-based ones, with some exceptions showing good performance.

Keywords: expert finding; information retrieval; machine learning; deep learning; recommender systems; academia; authorship attribution

Citation: de Campos, L.M.; Fernández-Luna, J.M.; Huete, J.F.; Ribadas-Pena, F.J.; Bolaños, N. Information Retrieval and Machine Learning Methods for Academic Expert Finding. *Algorithms* **2024**, *17*, 51. https://doi.org/10.3390/a17020051

Academic Editors: Edward Rolando Núñez-Valdez, Vicente García-Díaz and Frank Werner

Received: 18 December 2023
Revised: 11 January 2024
Accepted: 19 January 2024
Published: 23 January 2024

1. Introduction

In both our personal and professional lives, there are moments when our knowledge or ability to solve a problem falls short, necessitating us to connect with individuals whose experience or expertise can provide valuable assistance for the issue at hand. If a tool were available for such situations, it could assist us in locating these individuals, who may be represented within a system through various means, such as the services they offer (in structured or unstructured formats), CVs, blog entries, tags, and more. These matches would typically be identified based on our expressed needs, either through a natural language query or a simple list of keywords. This software would be responsible for determining the most suitable experts who can fulfill our requirements.

This problem is called expert finding and in general refers to the process of identifying individuals who possess specialized knowledge or expertise in a particular subject or field. The goal is to locate individuals who can provide valuable insight, guidance, or assistance in relation to a specific topic.

Expert finding has found success in various applications, including in relation to enterprises [1,2], community question answering [3,4], social networks [5,6], online forums [7], medicine [8], and academia [9–12]. A strong indicator of the significance of and interest in expert finding is the abundance of publications on the topic, including a substantial number of survey papers [13–19].

We focus on the academic or scholarly field, where the search for experts typically involves identifying the key figures (researchers, scholars, professors) who have pioneered fundamental concepts, made significant advancements, or possess in-depth knowledge and expertise in specific academic disciplines. Two illustrative use cases that exemplify the process of finding academic experts are as follows: (1) An international project office seeks to identify reviewers for evaluating project proposals submitted for a particular call. In this scenario, the office administrators aim to identify academics whose research expertise aligns with the proposal topics, as expressed through keywords, so they can carry out the evaluation task. (2) A multidisciplinary research team is engaged in a project spanning multiple knowledge domains, necessitating the inclusion of researchers from various fields. In such cases, the research team leaders strive to identify other research groups or specific researchers who excel in fields that make them essential for incorporation in their research.

Taking it to the next level, the goal of this paper is to study academic-expert-finding methods by utilizing information retrieval (IR) and machine learning (ML), including classical and deep learning models. In this case, the academics are represented by the papers published in journals or conferences, giving clear evidence of what their expertise fields are. Then, the recommendation of experts is carried out by providing a text where potential users express their information needs. These are typically represented by a kind of abstract describing the problem and the type of required knowledge. This might be seen as a recommendation process, where a hypothetical system recommends experts by means of ranking them, given an information need.

The IR-based methods construct multifaceted textual profiles (subprofiles) for each expert by clustering information from their scientific publications. An IR system (IRS) indexes these subprofiles and retrieves those most similar to the query expressing the required expertise. In this paper, we present a set of IR-based alternatives fully adapted to the problem. ML-based methods treat the problem as a classification task. They use the publications authored by each expert to train an automatic text classifier, where the categories represent different experts. We use different classifiers ranging from classical algorithms, like naive Bayes or logistic regression, to deep learning algorithms, such as convolutional or recurrent neural networks.

Hence, the two main objectives of this research are, firstly, to inquire into whether the concept of an author's profile could significantly contribute to the recommendation of academic experts and, secondly, to investigate whether there exists a technique drawn from IR and ML methods better suited for this specific task compared to others.

The contributions of this paper to the field of academic expert finding encompass the following key aspects:

- The development of information retrieval methods grounded in the concept of profiles, making it possible to capture the diverse ranges of topics covered by academic experts.
- The application of machine learning techniques to address this problem, an area that has received limited attention in previous research.
- A comprehensive performance evaluation comparing both types of methods. This evaluation was conducted through extensive experimentation using two collections of scientific publications: the PMSC-UGR dataset [20] and the CORD-19 dataset [21].

The remainder of our paper is organized as follows: Section 2 examines related work. Then, Section 3 provides an explanation of the two distinct approaches employed to address the expert-finding problem, and Section 4 delves into the specifics of the experiments conducted using the proposed methods, encompassing the experimental setup, the results obtained, and the subsequent discussion. Lastly, in Section 5, we conclude the paper by highlighting the primary findings of this work and suggesting potential avenues for further research in this domain.

2. Related Work

Within academia, in addition to works where the goal is to find experts for a general, unspecified task [11,22,23], there are also works focusing on more specific tasks, such as

supervisor identification [9] or reviewer recommendation for journals, conferences [12,24–27], or even grant proposals [28]. Another application involves searching for collaborators in a research field, which may entail seeking collaboration within the user's own research area or in a different field. These two situations may necessitate different methods. In the first case, when identifying experts with similar expertise, we can directly compute similarities between the target researcher and potential candidates [10,29]. In the second case, we need to identify candidates who match the terms or topics that describe the desired expertise [30,31]. In our case, we focus on general methods where the input to the system is text expressing the required knowledge.

Approaches to expert finding can typically be categorized as content-based models, link structure-based models (also known as network-based models), or hybrid models that combine elements of both [32]. Content-based models rely on textual content from documents associated with each expert for information [33–35]. In contrast, link structure-based models leverage scholarly networks based on co-authorship or co-citation relationships [36–39], often employing algorithms like PageRank, HITS, or random walk with restart to analyze these relationships. Since our models are content-based, our focus is on this type of approach.

Content-based models can utilize textual content directly, considering terms from documents. They often integrate all documents' text into a single personal profile or analyze each document separately [34,40–42]. In the first case, these models combine all the documents associated with each expert, merging their text into a single document that represents the expert's profile. This approach generates a unified profile for each expert that encompasses their expertise as demonstrated through their own documents. This method is called the single-document author model in [43] and also referred to as a candidate-centric or query-independent approach [44]. In the second case, each document associated with an expert is treated as an individual unit, forming a subprofile that focuses on the specific expertise related to the content of that document. With this method, an expert will have multiple subprofiles corresponding to the number of associated documents they have. This is known as a document-based method [33] and is also called a query-dependent approach [34].

If a single profile were constructed from all the documents of the experts, it would amalgamate all their topics of interest. This could lead to more general topics overshadowing specialized ones, resulting in an inaccurate representation of the expert's interests. Consequently, there is a risk that they might not be identified when searching for specific expertise. Conversely, the document-based method excessively atomizes the information about an expert's expertise by assigning one subprofile for each document, as many of the associated documents can be related to the same expertise. This extreme atomization could compromise the comprehensibility of the subprofiles. Regarding the efficacy of these two methods, while document-based approaches are generally deemed superior to profile-based methods [33], certain studies have arrived at the opposite conclusion [41,45]. Therefore, it may be a matter of the specific application being considered.

Our IR-based models adopt an intermediate approach by generating multiple subprofiles for each expert using clustering methods, aiming to ensure that each subprofile represents a specific topic or a homogeneous group of topics. Some related works also use clustering methods, such as [46,47], but they typically cluster people based on similar expertise, whereas our approach clusters documents to separate different kinds of expertise for each researcher. Our approach also shares similarities with the research in [43], which employed the author–persona–topic (APT) model to suggest suitable reviewers for submitted papers. In this model, authors can assume diverse "personas", each characterized by distinct distributions over hidden topics, representing various combinations of expertise. Similarly, our approach distributes a researcher's expertise into several subprofiles through clustering techniques.

Another avenue in content-based methods involves the use of topic models, such as LDA and its extensions [48–50], to represent content at a higher level than individual

terms [51,52]. In some of our IR-based models, we also employ LDA but with a different approach. We use LDA to discover hidden topics in the article collection and then cluster these articles by topics. This clustering process helps determine the subprofiles of each expert while retaining the representation of these subprofiles using terms, which may be an advantage regarding their expressiveness.

Another approach to manage content-based methods is by employing ML, particularly classification models, to rank experts based on models learned from their previously published articles. While relatively few works have explored this approach, some examples can be found in the field of community question answering (CQA). For instance, [53] used support vector machines (SVM) to detect different types of experts within the community. The authors of [54] utilized logistic regression to model the users' answer quality for previously answered questions, and [55] employed a binary SVM trained with question–user pairs to assess whether a given user could provide a satisfactory answer to a question. In the realm of deep learning models, [56] used recurrent neural networks for reviewer recommendation. In the CQA domain, other works have also leveraged deep learning techniques, such as convolutional neural networks [57,58], long short-term memory networks [59], and transformers [60]. Beyond CQA, deep learning models, especially convolutional neural networks, have found applications in financial problems [61].

3. Materials and Methods

In this section, we describe the two different approaches used to tackle the expert-finding problem.

3.1. Approaching the Problem Using Information Retrieval

From an IR perspective, the problem can be approached using the following generic strategy: we begin with a corpus of scientific articles authored by a group of researchers, from which we construct a collection of text documents that somehow reflect the researchers' interests, knowledge, and expertise. Each researcher is associated with either a single document/profile or multiple documents/subprofiles within this collection. For instance, when there are multiple documents, each one could be tied to a specific research interest of the researcher. The collection is subsequently indexed by an IRS. A query, representing the desired expertise, is then submitted to the IRS, which in turn provides a ranked list of profiles/subprofiles. To convert this list into a ranked list of recommended researchers, a method is applied that combines the scores of each subprofile associated with the same researcher.

In the previous section, we discussed the two primary methods for creating text documents that represent profiles or subprofiles: the candidate-centric and document-based models. These two approaches represent the two opposite ends of the spectrum: consolidating all the information about an author into a comprehensive, unified profile or fragmenting this information to the greatest extent possible, considering a single publication as the smallest unit of knowledge about a researcher.

An alternative, intermediate approach would be to group the publications of each researcher into clusters based on their thematic or topical similarities. This would result in multiple subprofiles for each researcher, reflecting the different topics covered in their work. By separating subprofiles based on the thematic coherence of articles, a more precise representation of each researcher's expertise can be achieved. Consequently, this method has the potential to improve the identification of suitable experts for specific information requirements. This approach is similar to the one which was proposed in [62] in the context of publication venue recommendation.

Our approach takes a global perspective, focusing on identifying the broader themes covered in our collection of scientific articles rather than discerning specific topics within an individual author's articles. Our goal is to capture the comprehensive topics that emerge from the entire corpus of articles, encompassing the works of all authors involved.

We can utilize a clustering algorithm to group all the articles within the corpus. This algorithm will categorize the articles into K clusters (K represents a parameter of the clustering algorithm).To derive the subprofiles associated with each author, we concatenate the text from all the articles written by that author that belong to the same cluster into a single document. Consequently, each author will have a maximum of K subprofiles. It is important to note that if an author has not published any articles on certain topics covered in a particular cluster, their corresponding subprofile for that cluster will be empty.

It should be noticed that when we approach the problem from an IR perspective, the fact that an article in the corpus may have several authors (which is quite common in this context) does not pose any technical problem: this article will be considered to build the profiles or one of the subprofiles of *each* of its authors.

3.2. Approaching the Problem Using Machine Learning

Another approach to tackling the problem is by employing machine learning methods, including classical or deep learning models. The process begins with a collection of articles written by a group of researchers. The problem is then framed as a supervised text classification task, where the classes represent different authors, the instances refer to the individual articles, and the features correspond to the words (textual content) present in these articles. Subsequently, a model can be trained to predict the most appropriate class when provided with a textual representation of the required expertise. By utilizing a model capable of assigning a score to each researcher, a ranked list of potential experts can be obtained.

However, when approaching the problem from a machine learning perspective, we encounter a technical issue stemming from the presence of multiple authors for articles in the corpus. In fact, our problem becomes one of multi-label classification since several labels (authors) could be associated with each instance. Therefore, we need to transform this multi-label classification problem into a single-label classification task. However, given the large number of possible authors (several thousands in the experiments below), the most common problem-transformation methods—namely, binary relevance (which involves learning one binary classifier for each class label) [63] and label powerset (which requires learning a single multi-class classifier, where each different label combination found in the multi-label data is a single label) [64]—can be prohibitive in our case.

Due to this rationale, we employ an alternative technique for problem transformation, as stated in [64]. This particular approach involves breaking down each training instance with multiple labels into separate instances, each with a single label. Subsequently, a single multi-class classifier is trained on this expanded dataset where the number of classes remains the same as the original problem.

We have used a number of very different classification models: from relatively simple models, such as naive Bayes, decision trees, K-nearest neighbors, logistic regression, and other linear classifiers [65], to more complex models, such as multi-layer perceptron, convolutional neural networks, long short-term memory networks, and bidirectional long short-term memory networks [66].

4. Experimental Results and Discussion

In this section, we provide details about the experiments carried out with the proposed methods, including the experimental setup and the obtained results, and discuss them.

4.1. Test Collections

We used two different collections of scientific publications to test the capabilities of the different methods. On the left-hand side of Figure 1, we present a summary of the datasets used and outline the procedure employed for partitioning the data into training and test sets.

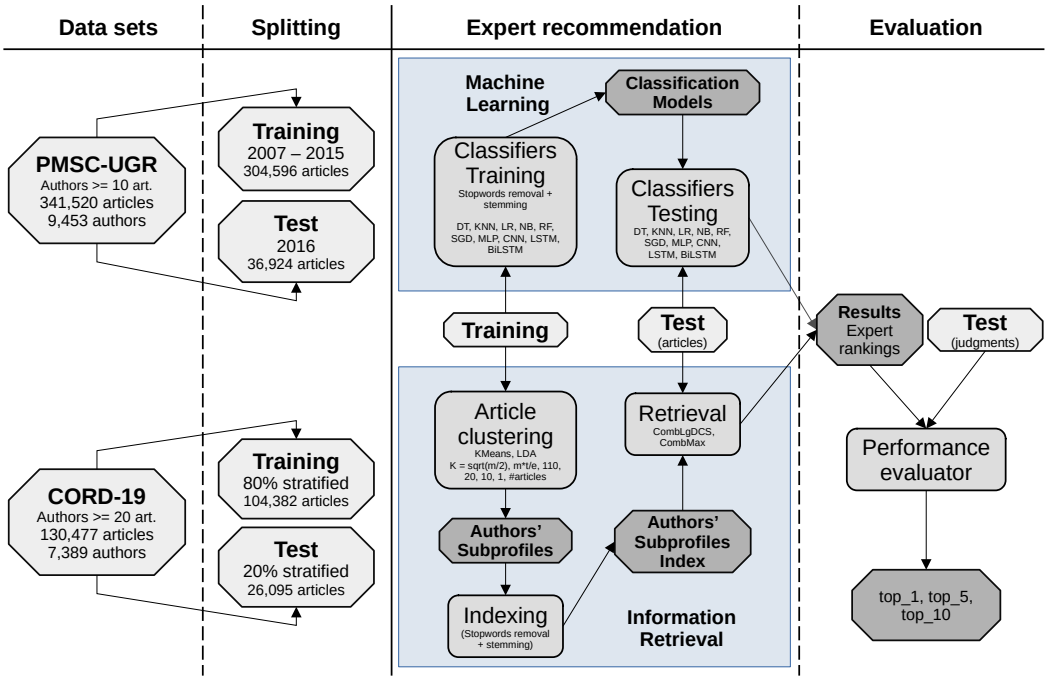

Figure 1. Graphical scheme of the experimental design: datasets and their partitions and ML- and IR-based recommendation and evaluation.

PMSC-UGR [20] is a collection of journal papers extracted from PubMed and Scopus focusing on the biomedical domain. One advantage of this collection is that the authors' names are disambiguated using their ORCID codes. Therefore, we avoid the issue of confusing publications between authors who share the same name. We specifically selected articles published between 2007 and 2016, utilizing the articles from 2007 to 2015 for the training set and those published in 2016 for the test set. Additionally, we only considered authors who had a minimum of 10 articles in the training set, ensuring a sufficient amount of information to train the models, and at least 1 article in the test set. As a result, this document collection consisted of 341,520 articles, with 304,596 articles in the training set and 36,924 articles in the test set, authored by 9453 authors.

The reason for this splitting was to prevent prediction of historical events based on future information; i.e., to refrain from suggesting potential experts based on the content of a test article by relying on information from articles published after the test article's date. To ensure this, the test articles, which represented the sought-after expertise, had to have a publication date later than the training articles used to form the experts' subprofiles. This criterion guided the division of the data into training and test sets.

The other collection was the COVID-19 open research dataset, CORD-19 [21], a resource for scholarly articles about COVID-19 and related coronaviruses (https://www.kaggle.com/datasets/allen-institute-for-ai/CORD-19-research-challenge, accessed 19 June 2023). In this dataset, authors' names are not disambiguated and we performed some basic procedures to clean data and deal with similar author names. To mitigate name ambiguity, the initial step involved filtering out entries in which authors had provided only their initials instead of their preferred canonical full names. For instance, in cases like "Smith L.", "Smith L", and "Smith Louis", only "Smith Louis" was recognized as a valid author. Furthermore, for enhanced reliability, we limited our analysis to authors with a minimum of 20 publications signed using their canonical full names, obtaining 7389 authors and 130,477 articles. We aimed to achieve a split between training and test partitions based on publication dates, following a similar approach as used for the PMSC-UGR collection. However, the concentration of articles within a brief time period posed a challenge for reli-

able date-based division. Additionally, editorial factors complicated the determination of whether an article published on a certain date genuinely followed another article published slightly earlier. As a result, we chose an alternative method for splitting: an 80/20 stratified sampling approach, obtaining 104,382 articles for training and 26,095 for test. Additionally, we guaranteed that a paper co-authored by multiple authors was not included in both the train and test splits simultaneously.

In Figure 2, we show two histograms that display the frequency for authors based on the number of published papers they had in each respective dataset. Despite employing distinct criteria concerning the time span and topics during the compilation of these collections, both exhibited a notable power-law pattern. This pattern revealed that a small fraction of authors had contributed significantly to the total number of papers, while the majority had made relatively fewer contributions, as expected. It is worth noting that a consistent pattern persisted when examining the dataset's division into training and test sets using two distinct splitting methods, as illustrated on the right-hand side of Figure 2, with a focus on the test partition. This observation suggested that employing either temporal splitting, as in PMSC-UGR, or stratified sampling, as in CORD-19, did not introduce bias into the distribution of the test set, thus giving robustness to the obtained results.

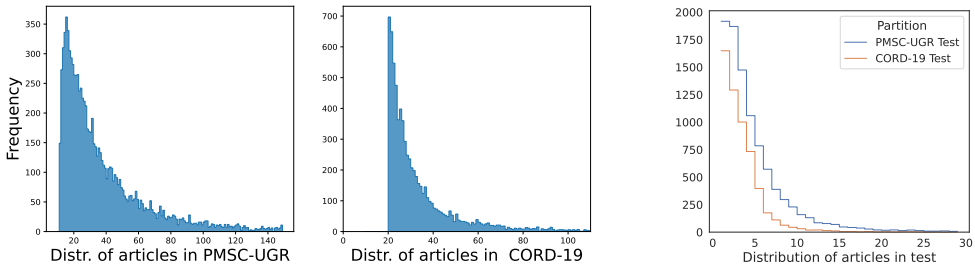

Figure 2. Author publication distribution in PMSC-UGR and CORD-19 datasets and articles' respective distributions in the test partitions.

Both collections contained textual information consisting of the title, abstract, and keywords for each article, as well as the authors' names/codes. This information was utilized differently depending on whether the article belonged to the training set or the test set. For training articles, the textual content was employed to train a model/profile that reflected the scientific interests and expertise of the authors. On the other hand, for an article in the test set, its content served the purpose of expressing the required information or expertise. It is important to note that the assumption underlying this approach was that the authors of the test articles were indeed experts in the topics discussed within them. This assumption was objective yet conservative, as it was reasonable to assume that other researchers working on the same or similar topics could also possess expertise in them. Consequently, our evaluation method resembled authorship attribution [67,68], in which the aim is to identify the authors of a test article (it is worth mentioning that we solely focus on the content of texts and do not employ stylometric features for this purpose, as is usual for authorship attribution [69]).

All of the methods considered take each test article as the input and use the learned model/profiles to obtain a list of potential experts; i.e., possible authors of the article (ranked by predicted scores). As an article may have several authors and we wanted to deal with each author separately, we duplicated those test articles with more than one author as many times as the number of authors. In this way, each test article was associated with only one author.

4.2. Evaluation Measures

The evaluation measures only aimed to detect the ability of the different models to identify the authors of the test articles. For that purpose, we used the top_n metric (called top_k_accuracy_score in scikit-learn or precision at k in [70]) for different values of n. This metric computes the number of times, in proportion, that the correct label (the true author)

is among the top n labels predicted. This metric also has relevance for problems where there are many possible labels for each instance but only a few labels are in fact associated with the instance (in our case, we had many possible authors for each article but only a very small fraction of them were the true authors, and we were not interested in counting the number of times that a researcher was correctly discarded as the author of an article). We used three values for top_n; namely, top_1 (which reflects the number of times that a true author of an article is the first recommended expert), top_5, and top_10. Note that top_1 is interesting as knowing one of the authors is usually enough to know which group or lab a work comes from, which is an important input for our research task. Given that we associated each test article with only one author, this implies that several metrics were collapsed in this case: top_1 = accuracy = ndcg@1 = averagePrecision@1. Furthermore, top_k = recall@k = precision@k. Therefore, we believe that employing multiple instances of top_k captures the fundamental information about the system's performance, which is the reason these were the evaluation measures used in this research. In Figure 1, the evaluation process is depicted on the right-hand side.

4.3. Experimental Setup for the IR Models

In this section, we provide the technical details about the experiments with the IR-based approaches, as well as the software tools and parameters used for both clustering and retrieval. This is additionally illustrated in the lower section of the "Expert recommendation" column in Figure 1, showcasing the information retrieval (IR) workflow.

4.3.1. Clustering

To construct the term subprofiles of each researcher, we leveraged the article collections using two methods. Firstly, we utilized the K-means clustering algorithm [71]. Secondly, we employed latent Dirichlet allocation (LDA) [48] to generate K latent topics and subsequently performed clustering by assigning each article to its most probable topic. For the implementation of K-means, we relied on the scikit-learn Python library (https://scikit-learn.org, accessed 19 June 2023), whereas for LDA, we utilized the Python implementation available in the Gensim library (https://pypi.org/project/gensim, accessed 19 June 2023).

To address the large size of the document-term matrices in both collections (consisting of hundreds of thousands of rows representing articles and columns representing different terms), we employed a dimensionality reduction process. After removing stopwords and applying stemming, we disregarded terms that appeared in fewer than 750 articles or more than 90% of the articles. Additionally, to further reduce the dimensionality, we selected a maximum of 5000 of the most frequent remaining terms in the corpus as input for both K-means and LDA. However, it is important to note that the subprofiles derived from these algorithms encompassed all the terms present in the original articles once the clusters were generated.

Regarding the determination of the number of clusters, K, we conducted experiments using various approaches. Two of them were derived from conventional cluster analysis methods, while in the other three cases the value of K was fixed:

- We employed the formula $K = \sqrt{m/2}$ [72], where m represents the number of articles in the training set. Consequently, for PMSC-UGR and CORD-19, we obtained K values of 390 and 239, respectively.
- Another approach we considered was $K = m * t/e$ [71], where t corresponds to the number of distinct terms appearing in the articles within the training set, and e represents the number of non-zero entries in the document-term matrix. For PMSC-UGR and CORD-19, we obtained K values of 54 and 29, respectively.
- We set $K = 110$, which aligned with the number of descriptors or categories in the second level of the Medical Subject Headings (MeSH) thesaurus (https://meshb.nlm.nih.gov/treeView, accessed 4 July 2021).

- We set $K = 20$, the number of medical specialties at St. George's University (https://www.sgu.edu/blog/medical/ultimate-list-of-medical-specialties, accessed 4 July 2021).
- We set $K = 10$ to test a case with a very low number of clusters.

We also experimented with two extreme cases where no explicit clustering was performed: one where all the articles of a given author were grouped into a single profile (equivalent to using $K = 1$), which we called the Joint model, and another where each article of an author formed an independent subprofile (equivalent to using $K =$ articles in the collection), which was called the Atomic model.

4.3.2. IR Model

After obtaining the clusters (utilizing either K-means or LDA with a specific value of K), we proceeded to generate the corresponding document collection that represented the subprofiles linked to each author. To facilitate indexing, we employed the Lucene library (https://lucene.apache.org, accessed 15 January 2023) after eliminating stopwords and performing stemming. For IR, we opted for the Language Model as our chosen model, specifically utilizing the default Jelinek–Mercer smoothing implementation within Lucene.

We created a query for each article in the test set by using its textual content and submitted it to the IRS. The IRS then provided us with a ranked list of authors' subprofiles as the outcome. Since our desired output was a ranking of authors, we had to perform a final fusion step to combine the scores of all the subprofiles associated with the same author. This step ensured that the authors were appropriately ranked. There are several options available for this fusion purpose [73].

We tested two fusion strategies. The first method was called *CombLgDCS* [74]. Essentially, this approach sums up the scores of all the subprofiles associated with the same author, considering only those that appear in the first p positions of the ranking and applying logarithmic devaluation based on their positions. In the second method, named *CombMax*, the author's score is determined by their highest-scoring subprofile.

The results were very similar, with a slight advantage for *CombMax* when used with PMSC-UGR (in this context, it seems that the subprofile most closely aligned with the query captured the relevant context, while the other subprofiles contributed only marginally) and a slightly better performance for *CombLgDCS* when applied to CORD-19. We believe that this behavior was due to the characteristics of the document collections because PMSC-UGR covers a broader range of topics while CORD-19 is focused on the study of COVID-19. This makes the themes in the latter more interconnected, and it is beneficial to combine the relevance of various aspects involved in a specific query. We therefor present the best results obtained with each dataset accordingly.

4.4. Experimental Setup for the ML Models

We now present the technical information about the experiments conducted using ML-based approaches. This information is summarized in the top part of the "Expert recommendation" column in Figure 1. Specifically, we utilized the scikit-learn and Keras libraries [75]. Scikit-learn (https://scikit-learn.org, accessed 15 January 2023) is an open-source ML library designed for Python, while Keras (https://keras.io, accessed 15 January 2023) is a Python-based deep learning API that runs on top of the TensorFlow machine learning platform. The scikit-learn library was applied to classical models, whereas the Keras library was employed for deep learning models, which are not available in the scikit-learn library.

A total of 10 ML models divided into two categories were utilized. On one hand, we employed decision trees (DTs), K-nearest neighbors (KNN), logistic regression (LR), naive Bayes (NB), random forests (RFs), and stochastic gradient descent (SGD) linear models. On the other hand, we utilized multi-layer perceptron (MLP), convolutional neural networks (CNNs), long short-term memory (LSTM), and bidirectional long short-term memory (BiLSTM).

We conducted numerous experiments and carefully selected a subset for reporting. For instance, we experimented with using the text of the articles as input data, both with and without feature selection. The feature selection involved removing terms that appeared in fewer than 750 articles or more than 90% of the articles, following the same selection criteria as used by the clustering algorithms; stop-word removal and stemming were always applied, regardless of whether feature selection was used or not. Additionally, we explored the conversion of the text into numerical vectors using two transformer-based architectures: bidirectional encoder representations from transformers (BERT) [76], which uses a pre-trained model with PubMed documents to extract the vector associated with the `[CLS]` special token as a contextual representation of the whole text, and a sentence-BERT (SBERT) model [77], which uses the fine-tuned model `sentence-transformers/allenai-specter` consisting of a fine-tuned pooling layer on top of a transformer encoder.

However, our findings indicated that experiments using feature selection consistently yielded worse results compared to those without it. Therefore, we only report the results obtained without feature selection. Similarly, the experiments using BERT consistently performed worse than those using SBERT, so we solely present the results using SBERT. Moreover, we explored various configurations for some of the models, such as different numbers of layers and neurons in each layer for deep learning models. However, we only report the results of the best configuration for each model that we employed.

Concerning the configuration parameters of the different classical models, we used all the default values for the DT, KNN, LR, and NB models. For the RF model, we set `max_depth=20`, which represents the maximum depth of the tree. Regarding SGD, we employed `loss='modified_huber'` because it allows probability estimates, which are not available for other loss functions.

For the deep learning models, the configuration parameters were as follows:

- For MLP, we used only one hidden layer with 8500 neurons for PMSC-UGR and 5000 for CORD-19 and trained for 20 epochs.
- For the CNN, LSTM, and BiLSTM models, we employed an embedding layer with pre-trained weights obtained from PubMed with a sequence length of 1024, an embedding size of 200, and 10 iterations.
- Additionally, for the CNN, we utilized one convolutional layer consisting of 2048 neurons, a kernel size of 3, and a stride of 1; for the LSTM, we used two LSTM layers, each with 200 neurons; lastly, for the BiLSTM, we employed two BiLSTM layers, one with 200 neurons and the other with 100 neurons.

4.5. Results and Discussion

In this section, we present and discuss the results of the experiments with the two document collections. Before diving into the details, it is worth pointing out that these experiments were carried out with an Intel(R) Xeon(R) Gold 6230 CPU @ 2.10 GHz with 20 cores, 314 GB of RAM, and 14 TB of space in the HDD running a Fedora 39 distribution.

Table 1 presents the results obtained with the IR-based models for the PMSC-UGR collection. The results exhibit a notable degree of consistency, with only minor variations when a different number of clusters, K, was used. This observation is particularly interesting, as it suggests that the choice of this parameter is not of critical importance. Furthermore, the results consistently favored K-means over LDA, although the distinctions were more pronounced for top_5 and top_10 than for top_1.

Table 1. Top_n results obtained for PMSC-UGR with the IR-based models.

K	top_1	top_5	top_10	top_1	top_5	top_10
		Using K-means			Using LDA	
10	0.3566	0.5874	0.6644	0.3558	0.5527	0.6316
20	0.3565	0.5861	0.6647	0.3555	0.5585	0.6358
54	0.3596	0.5863	0.6623	0.3563	0.5617	0.6405
110	0.3587	0.5855	0.6587	0.3564	0.5645	0.6434
390	0.3580	0.5812	0.6566	0.3580	0.5692	0.6454
Atomic	0.3502	0.5706	0.6453			
Joint	0.3518	0.5847	0.6659			

When focusing on K-means, it became apparent that the inclusion of subprofiles enhanced the performance of extreme alternatives (namely, Atomic and Joint), especially for top_1 and top_5, provided that *K* was not excessively high. For top_10, a relatively low value for *K* was preferable, and in this scenario, the best outcome was achieved when using a single profile per author.

Remarkably, the system demonstrated an ability to correctly identify the author or expert in more than one third of the articles, and in over two thirds of cases, the correct author was found within the first 10 recommended authors out of a pool of 9453 potential authors.

Table 2 presents the results of our experiments with ML models for PMSC-UGR. It is noteworthy that some of the classical models, such as the DT, KNN and NB models, produced relatively poor results when applied to the text data. However, there was a significant improvement in their performance when we used SBERT embeddings instead of the original text, except for the DT model. On the other hand, models like SGD, LR, and MLP exhibited better results when using the raw text data.

Table 2. Top_n results obtained for PMSC-UGR with the ML-based models.

Model	top_1	top_5	top_10	top_1	top_5	top_10
		Using text			Using SBERT	
DT	0.0537	0.0613	0.0622	0.0227	0.0501	0.0703
NB	0.0665	0.1438	0.1910	0.1681	0.3641	0.4615
KNN	0.0683	0.1620	0.1624	0.2428	0.4703	0.4706
RF	-	-	-	0.1181	0.2344	0.3002
SGD	0.2272	0.4188	0.4839	0.1785	0.2648	0.2667
LR	0.1948	0.3696	0.4470	0.2707	0.5010	0.5918
MLP	0.3441	0.5791	0.6621	0.2365	0.4600	0.5520
CNN	0.1425	0.2888	0.3600			
LSTM	0.0821	0.2196	0.3111			
BiLSTM	0.0894	0.2417	0.3331			

Interestingly, LR showed substantial improvements when SBERT embeddings were utilized instead of the text, while models like SGD and MLP displayed the opposite behavior. Among the deep learning models, including the CNN, LSTM and BiLSTM models, the results were consistently worse than those obtained with SGD, LR and MLP. In absolute terms, the best results were achieved by MLP using the raw text data.

Unlike the results of the IR-based models, the ML-based ones exhibited considerable variability, spanning from very low to moderately good performance (see Figure 3). However, it is worth noting that even the best results achieved by the ML-based models were somewhat inferior to those of the IR-based models.

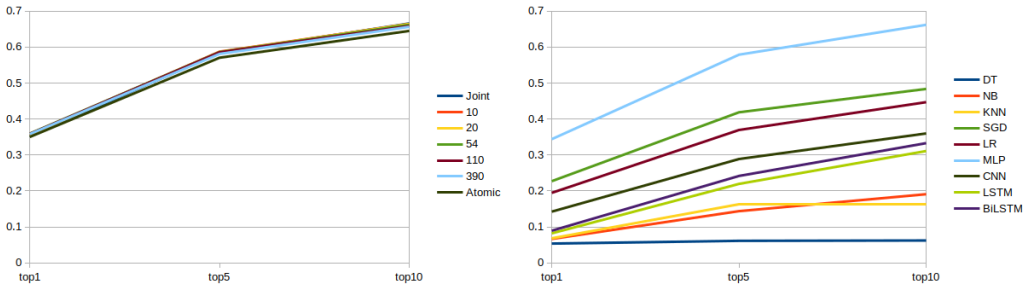

Figure 3. Top_n results for PMSC-UGR for the models based on IR using K-means (**left**) and ML using text (**right**).

Tables 3 and 4 display the results obtained from the CORD-19 collection using IR-based and ML-based models, respectively. We can observe that the obtained results partially exhibited similar trends to those of the previous experiments with PMSC-UGR while also revealing some distinct patterns.

In the case of the IR-based models, the results continued to demonstrate a significant degree of similarity, even more pronounced than in the other dataset. Remarkably, neither K-means nor LDA showed superior performance compared to the other in this context. Additionally, it appears that the results slightly improved when a larger number of clusters was utilized.

Table 3. Top_n results obtained for CORD-19 with the IR-based models.

K	top_1	top_5	top_10	top_1	top_5	top_10
		Using K-means			Using LDA	
10	0.1760	0.2812	0.3228	0.1717	0.2791	0.3218
20	0.1792	0.2823	0.3244	0.1779	0.2813	0.3234
29	0.1778	0.2834	0.3240	0.1787	0.2840	0.3248
110	0.1861	0.2893	0.3285	0.1853	0.2922	0.3316
239	0.1901	0.2919	0.3279	0.1883	0.2928	0.3313
Atomic	0.2006	0.3047	0.3466			
Joint	0.1717	0.2874	0.3351			

Table 4. Top_n results obtained for CORD-19 with the ML-based models.

Model	top_1	top_5	top_10	top_1	top_5	top_10
		Using text			Using SBERT	
DT	0.0602	0.0745	0.0758	0.0093	0.0239	0.0359
NB	0.0166	0.0372	0.0504	-	-	-
KNN	0.0389	0.1304	0.1308	0.0835	0.2241	0.2245
RF	-	-	-	0.0363	0.0714	0.0929
SGD	0.1308	0.2397	0.2798	0.0614	0.0967	0.0983
LR	0.0638	0.1275	0.1659	0.1084	0.2162	0.2704
MLP	0.1852	0.3121	0.3696	0.0954	0.1939	0.2401
CNN	0.0486	0.1097	0.1495			
LSTM	0.0362	0.0927	0.1322			
BiLSTM	0.0401	0.1006	0.1411			

Among the ML-based models, the DT, NB, and KNN models continued to exhibit consistently poor results. However, it is worth noting that KNN showed significant improvement when SBERT embeddings were utilized. On the other hand, SGD, LR, and MLP demonstrated better performance. Furthermore, LR's performance still benefited from the use of SBERT embeddings, while the SGD and MLP models' performance declined in this scenario. The deep learning algorithms (CNN, LSTM, and BiLSTM) fell somewhere in

between, with MLP still being the standout performer, achieving even better results than the IR-based models.

It is also important to highlight that the absolute results in this case were considerably lower than those obtained with PMSC-UGR, showing an average reduction in performance of approximately 50%. One possible explanation for this discrepancy is that CORD-19 is a more homogeneous collection centered around COVID-19, while PMSC-UGR covers a broader range of topics within biomedicine (just to illustrate this fact, we can highlight that the number of unique terms (i.e., the vocabulary size) in PMSC-UGR is 506,536, whereas in CORD-19, it is 180,501). Consequently, distinguishing between authors based solely on textual content was more challenging within the CORD-19 dataset.

Finally, and in terms of efficiency, we have to mention that the IR methods were faster during training compared to neural network-based approaches, as the training phase was considerably longer for the latter, especially when compared to the construction and indexing of profiles for IR techniques. As an example, for those experiments with the best performance, the clustering with K = 110 and the corresponding indexing for the subprofiles obtained from PMSC-UGR lasted 8529″ (2.36 h), while for CORD-19 and K = 239, they lasted 7060″ (1.96 h). In contrast, the training phase for MLP was 163,478″ (45.41 h) for PMSC-UGR and 54,137″ (15.04 h) for CORD-19. The model construction times from the IR models represented 5% of the training time for the first dataset and 13% for the second with respect to MLP. While retrieval and classification times could be comparable for both types of techniques once the models are trained (less than 0.2 s per test article), ML-based methods are less suitable for real-time environments with high dynamism, such as in this case where new authors and papers continuously arrive in the system. Maintaining document collection freshness in ML requires frequent model updates, which can be resource-intensive. This is not as crucial for clustering and indexing in IR-based models, which can be updated rapidly.

5. Conclusions

In this paper, we addressed the challenge of academic expert finding; i.e., the task of searching for researchers who are experts in a particular scientific field. We approached this problem using two distinct methods: information retrieval and machine learning. Our objective was to compare these techniques from these two major areas and determine if one was more suitable than the other in terms of performance for this problem.

On one hand, IR-based techniques considered all the published papers of the authors as a collection of documents. These documents were used to feed an IRS, which generated a ranking of authors based on the provided information needs. More specifically, IR methods were grounded in the concept of profile, which acted as a "container" for storing an author's papers, thereby reflecting their expertise. We explored three types of profiles, ranging from the simplest approaches to more sophisticates ones: (1) a single profile containing all the text from an author's published papers; (2) treating each paper as an individual subprofile, resulting in a collection of subprofiles within the overall profile; and (3) different subprofiles based on distinct clusters/topics mined from the whole dataset. To achieve this, we employed methods such as clustering and LDA to extract and capture the underlying topics from the entire document collection and build subprofiles for each author with papers from the same topic, trying to acquire the different topics in which the author was an expert.

On the other hand, ML-based techniques treated this problem as a classification task. In this approach, publications and their associated words served as features, while their corresponding authors were treated as classes. In this study, we conducted an extensive evaluation of classifiers spanning from classical methods to the latest advancements in deep learning. To the best of our knowledge, there is a noticeable absence of prior comprehensive experimental studies, despite the apparent potential, applying ML and deep learning techniques to the task of expert finding. This fact is configured as one of the contributions of this paper.

To assess the effectiveness of the various techniques employed in this study, we subjected them to testing with two distinct collections of scientific papers: PMSC-UGR and CORD-19. The former comprises published articles in the wider field of biomedicine, while the latter focuses more specifically on heterogeneous aspects related to COVID-19. These datasets also exhibit disparities in size, including the numbers of papers and authors, as well as variations in the homogeneity of topics.

When considering the results obtained during the evaluation phase, it is important to emphasize that IR techniques were well suited for this problem, consistently delivering strong performance. They possessed a significant advantage in that they yielded more consistent results across various techniques, making their application less reliant on parameter-tuning (e.g., the number of clusters/topics used) and computational resources. In contrast, ML methods generally exhibited lower performance, with the exception of MLP, which achieved results similar to those of IR-based techniques but required careful parameter optimization.

When we delve into the insights garnered from IR-based techniques, specifically those centered around topicality, such as clustering using K-means and LDA, we find that they performed similarly to the more basic Joint and Atomic models, albeit with marginal differences. However, their application could prove beneficial in real-world search scenarios where the information derived from clusters can provide added value by offering explanations for recommendations. For instance, in a study such as the one mentioned in [78] where an evaluation with real users was conducted, tag clouds from various subprofiles played a pivotal role as explanatory tools for the recommender system's outputs. Another potential advantage of utilizing profiles is that, by having author–topic pairs as the information source prior to the ultimate fusion of the retrieval model, it becomes feasible to suggest groups of experts corresponding to each of the primary topics encompassed within the required expertise.

When comparing various ML algorithms, one notable observation is that the state-of-the-art deep learning techniques utilized in this study did not consistently outperform the other classification algorithms. However, it is worth highlighting the commendable performance of the MLP classifier, which is also rooted in neural networks. A plausible explanation for this phenomenon could be that, despite the datasets used in the experimentation being of substantial size, they might need to be even larger to fully leverage the capabilities of deep learning methods, particularly considering the very large number of authors (classes) involved in the experimentation. Another hypothesis that could explain the limitations of transformer-based approaches is that BERT and SBERT provide semantic representations for entire documents. However, in some instances, expertise attribution may be linked to a specific topic within a text that discusses various subjects. IR-based methods appear to be better equipped to manage such associations compared to the single representations of entire documents provided by BERT and SBERT.

Regarding the dual research question presented in Section 1, it is worth noting that the widespread utilization of profiles contributed moderately to enhancing the resolution of the problem. These structures have the capacity to effectively encapsulate the core subjects in which authors possess expertise. This enhancement was quantifiably demonstrated by the slightly superior performance observed in our experiments. However, the true potential lies in the realm of explanation, as previously highlighted. Concerning the comparison between IR and ML for the specific problem under consideration, we can conclude that IR outperforms ML in most cases, with the exception of a specific ML technique, as stated in the preceding paragraph.

While this study offers valuable insights into the academic-expert-finding problem, it is crucial to acknowledge certain limitations that warrant further investigation. Firstly, in this research only biomedicine-related datasets were used. Extending the analysis to other knowledge domains would help assess the models' performance across diverse fields. Secondly, the ML algorithms were mostly executed with default parameters. Exploring optimal parameter values could enhance the models' effectiveness. Moreover, the IR segment

of the study examined only one retrieval model. Evaluating alternative models would contribute to a more comprehensive understanding of the information retrieval process. Finally, the IR and ML models were implemented and tested in a research environment, focusing solely on content. Relevant features for a production system, such as bibliometric indexes or temporality, were intentionally not incorporated. In such cases, the output of our model could serve as input for various re-ranking strategies. These identified limitations underscore the necessity for future research to delve into specific aspects within a broader and more varied context.

The primary practical implication drawn from this in-depth empirical study, assuming the implementation of these models in a live production environment for a real-time academic recommendation system, is that while the top-performing IR and ML models exhibit similar performance, we would advocate for an IR-based solution. The key rationale behind this preference lies in the dynamic nature of the problem, where new authors and papers frequently enter the recommender system. The construction of IR models proved to be faster compared to the training of ML models, particularly those based on neural networks. Additionally, quick methods can be devised to seamlessly integrate new authors and their papers into the IR-based system without the need to rebuild data structures from scratch. This ensures that the system is continually updated. In contrast, ML models in this scenario would require retraining with all available data, which, while feasible offline, sacrifices the system's real-time freshness.

In addition to these previous ideas, and also in terms of future research directions, this study gives rise to several further promising avenues. The first involves the integration of supplementary information extracted from the document collections into the models, including scholarly co-authorship and citation networks. The underlying assumption here is that these additional data, which establish relationships among authors and their associated works, could prove highly beneficial in enhancing the models' performance. This is because the models would not solely rely on textual data but would also be enriched with information that could bolster the evidence of an author's expertise, as discussed in previous studies [32,79].

In this study, we recommended experts in a specific topic even if they actively worked in that area 20 years ago, provided that their historical work aligned with the current information need. However, it may be advantageous to prioritize experts whose publications in that same topic are more recent, as this suggests they are more likely to be up-to-date. This consideration prompts us to explore adaptations of the methods developed in [80] to accommodate the possibility of researchers' expertise evolving over time. Such adaptations would involve assigning greater weight to their more recent expertise, aligning with the dynamic nature of their knowledge.

Lastly, we are in the process of developing a practical search application that empowers users to discover academic experts for a wide range of purposes, including finding collaborators for research, paper reviewers, and project evaluators, among others. This application will incorporate explanatory features, entirely reliant on the models, aimed at enhancing user understanding of the recommendations. These enhancements will be designed to strengthen user trust and overall satisfaction with the recommender system.

Author Contributions: Conceptualization, L.M.d.C., J.M.F.-L. and J.F.H.; methodology, L.M.d.C., J.M.F.-L. and J.F.H.; software, N.B., J.M.F.-L., J.F.H. and F.J.R.-P.; validation, N.B., J.M.F.-L., J.F.H. and F.J.R.-P.; formal analysis, L.M.d.C., J.M.F.-L. and J.F.H.; investigation, N.B., L.M.d.C., J.M.F.-L., J.F.H. and F.J.R.-P.; resources, L.M.d.C., J.M.F.-L., J.F.H. and F.J.R.-P.; data curation, N.B., J.M.F.-L. and J.F.H.; writing—original draft preparation, N.B., L.M.d.C., J.M.F.-L. and J.F.H.; writing—review and editing, N.B., L.M.d.C., J.M.F.-L., J.F.H. and F.J.R.-P.; visualization, L.M.d.C., J.M.F.-L. and J.F.H.; supervision, L.M.d.C., J.M.F.-L. and J.F.H.; project administration, L.M.d.C., J.M.F.-L. and J.F.H.; funding acquisition, J.M.F.-L., J.F.H. and F.J.R.-P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the Spanish "Agencia Estatal de Investigación" under grants PID2019-106758GB-C31 and PID2020-113230RB-C22, in part by the Spanish "FEDER/Junta de

Andalucía-Consejería de Transformación Económica, Industria, Conocimiento y Universidades" under grant A-TIC-146-UGR20, and in part by the European Regional Development Fund (ERDF-FEDER).

Data Availability Statement: Publicly available datasets were analyzed in this study. PMSC-UGR can be found at https://utai.ugr.es/en/research/resources (accessed on 18 December 2023), and CORD-19 can be found at https://www.kaggle.com/datasets/allen-institute-for-ai/CORD-19-research-challenge (accessed on 18 December 2023).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Ruiz-Martínez, J.M.; narro-Giménez, J.A.M.; Martínez-Béjar, R. An ontological model for managing professional expertise. *Knowl. Manag. Res. Pract.* **2016**, *14*, 390–400. [CrossRef]
2. Alhabashneh, O.; Iqbal, R.; Doctor, F.; James, A. Fuzzy rule based profiling approach for enterprise information seeking and retrieval. *Inf. Sci.* **2017**, *394*, 18–37. [CrossRef]
3. Neshati, M.; Fallahnejad, Z.; Beigy, H. On dynamicity of expert finding in community question answering. *Inf. Process. Manag.* **2017**, *53*, 1026–1042. [CrossRef]
4. Xu, C.; Wang, X.; Guo, Y. Collaborative Expert Recommendation for Community-Based Question Answering. In *European Conference, ECML PKDD 2016, Machine Learning and Knowledge Discovery in Databases*; Lecture Notes in Artificial Intelligence; Frasconi, P., Landwehr, N., Manco, G., Vreeken, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9851, pp. 378–393.
5. Bozzon, A.; Brambilla, M.; Ceri, S.; Silvestri, M.; Vesci, G. Choosing the right crowd: Expert finding in social networks. In Proceedings of the 16th International Conference on Extending Database Technology, Genoa, Italy, 18–22 March 2013; pp. 637–648.
6. Xie, X.; Li, Y.; Zhang, Z.; Pan, H.; Han, S. A topic-specific contextual expert finding method in social network. In Proceedings of the Asia-Pacific Web Conference, Suzhou, China, 23–25 September 2016; pp. 292–303.
7. Omidvar, A.; Garakani, M.; Safarpour, H.R. Context based user ranking in forums for expert finding using wordnet dictionary and social network analysis. *Inf. Technol. Manag.* **2014**, *15*, 51–63. [CrossRef]
8. Tekin, C.; Atan, O.; Schaar, M.V.D. Discover the expert: Context-adaptive expert selection for medical diagnosis. *IEEE Trans. Emerg. Top. Comput.* **2015**, *3*, 220–234. [CrossRef]
9. Alarfaj, F.; Kruschwitz, U.; Hunter, D.; Fox, C. Finding the right supervisor: Expert-finding in a university domain. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, Montreal, QC, Canada, 3–8 June 2012; pp. 1–6.
10. Gollapalli, S.D.; Mitra, P.; Giles, C.L. Similar researcher search in academic environments. In Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries, Washington, DC, USA, 10–14 June 2012; pp. 167–170.
11. Cifariello, P.; Ferragina, P.; Ponza, M. Wiser: A semantic approach for expert finding in academia based on entity linking. *Inf. Syst.* **2019**, *82*, 1–16. [CrossRef]
12. Ishag, M.I.M.; Park, K.H.; Lee, J.Y.; Ryu, K.H. A pattern-based academic reviewer recommendation combining author-paper and diversity metrics. *IEEE Access* **2019**, *7*, 16460–16475. [CrossRef]
13. Lin, S.; Hong, W.; Wang, D.; Li, T. A survey on expert finding techniques. *J. Intell. Inf. Syst.* **2017**, *49*, 255–279. [CrossRef]
14. Al-Taie, M.Z.; Kadry, S.; Obasa, A.I. Understanding expert finding systems: Domains and techniques. *Soc. Netw. Anal. Min.* **2018**, *8*, 57. [CrossRef]
15. Gonçalves, R.; Dorneles, C.F. Automated expertise retrieval: A taxonomy-based survey and open issues. *Acm Comput. Surv.* **2019**, *52*, 1–30. [CrossRef]
16. Husain, O.; Salim, N.; Alias, R.A.; Abdelsalam, S.; Hassan, A. Expert finding systems: A systematic review. *Appl. Sci.* **2019**, *9*, 4250. [CrossRef]
17. Yang, Z.; Liu, Q.; Sun, B.; Zhao, X. Expert recommendation in community question answering: A review and future direction. *Int. J. Crowd Sci.* **2019**, *3*, 348–372. [CrossRef]
18. Yuan, S.; Zhang, Y.; Tang, J.; Hall, W.; Cabota, J.B. Expert finding in community question answering: A review. *Artif. Intell. Rev.* **2020**, *53*, 843–874. [CrossRef]
19. Zhang, Z.; Patra, B.G.; Yaseen, A.; Zhu, J.; Sabharwal, R.; Roberts, K.; Cao, T.; Wu, H. Scholarly recommendation systems: A literature survey. *Knowl. Inf. Syst.* **2023**, *65*, 4433–4478. [CrossRef]
20. Albusac, C.; de Campos, L.M.; Fernández-Luna, J.M.; Huete, J.F. PMSC-UGR: A test collection for expert recommendation based on PubMed and Scopus. In *Advances in Artificial Intelligence, CAEPIA 2018*; Lecture Notes in Artificial Intelligence; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11160, pp. 34–43.
21. Wang, L.L.; Lo, K.; Chandrasekhar, Y. CORD-19: The Covid-19 Open Research Dataset. *arXiv* **2020**, arXiv:2004.10706.
22. Moreira, C.; Wichert, A. Finding academic experts on a multisensor approach using Shannon's entropy. *Expert Syst. Appl.* **2013**, *40*, 5740–5754. [CrossRef]
23. Smirnova, E.; Balog, K. A user-oriented model for expert finding. In *Advances in Information Retrieval, ECIR 2011*; Lecture Notes in Computer Science; Clough, P., Ed.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6611, pp. 580–592.

24. Liu, D.; Xu, W.; Du, W.; Wang, F. How to choose appropriate experts for peer review: An intelligent recommendation method in a big data context. *Data Sci. J.* **2015**, *14*, 16. [CrossRef]
25. Tran, H.D.; Cabanac, G.; Hubert, G. Expert suggestion for conference program committees. In Proceedings of the 11th International Conference on Research Challenges in Information Science, Brighton, UK, 10–12 May 2017; pp. 221–232.
26. Zhao, S.; Zhang, D.; Duan, Z.; Chen, J.; Zhang, Y.; Tang, J. A novel classification method for paper-reviewer recommendation. *Scientometrics* **2018**, *115*, 1293–1313. [CrossRef]
27. Medakene, A.N.; Bouanane, K.; Eddoud, M.A. A new approach for computing the matching degree in the paper-to-reviewer assignment problem. In Proceedings of the 2019 International Conference on Theoretical and Applicative Aspects of Computer Science (ICTAACS), Skikda, Algeria, 15–16 December 2019; Volume 1, pp. 1–8.
28. Hettich, S.; Pazzani, M.J. Mining for proposal reviewers: Lessons learned at the national science foundation. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 August 2006; pp. 862–871.
29. Kong, X.; Mao, M.; Liu, J.; Xu, B.; Huang, R.; Jin, Q. Tnerec: Topic-aware network embedding for scientific collaborator recommendation. In Proceedings of the 2018 IEEE Smartworld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovation (Smart-World/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Grenoble, France, 7–11 October 2018; pp. 1007–1014.
30. Araki, M.; Katsurai, M.; Ohmukai, I.; Takeda, H. Interdisciplinary collaborator recommendation based on research content similarity. *IEICE Trans. Inf. Syst.* **2017**, *100*, 1–8. [CrossRef]
31. Cohen, S.; Ebel, L. Recommending collaborators using keywords. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 959–962.
32. Lin, L.; Xu, Z.; Ding, Y.; Liu, X. Finding topic-level experts in scholarly networks. *Scientometrics* **2013**, *97*, 797–819. [CrossRef]
33. Balog, K.; Fang, Y.; de Rijke, M.; Serdyukov, P.; Si, L. Expertise retrieval. *Found. Trends Inf. Retr.* **2012**, *6*, 127–256. [CrossRef]
34. Petkova, D.; Croft, W.B. Hierarchical language models for expert finding in enterprise corpora. In Proceedings of the IEEE International Conference on Tools with Artificial Intelligence, Washington, DC, USA, 13–15 November 2006; pp. 599–608.
35. Javadi, S.; Safa, R.; Azizi, M.; Mirroshandel, S.A. A recommendation system for finding experts in online scientific communities. *J. Data Min.* **2020**, *8*, 573–584.
36. Liu, X.; Bollen, J.; Nelson, M.L.; Sompel, H.V. Co-authorship networks in the digital library research community. *Inf. Process. Manag.* **2005**, *41*, 1462–1480. [CrossRef]
37. Ding, Y.; Yan, E.; Frazho, A.; Caverlee, J. PageRank for ranking authors in co-citation networks. *J. Am. Soc. Inf. Sci. Technol.* **2010**, *60*, 2229–2243. [CrossRef]
38. Yan, E.; Ding, Y. Discovering author impact: A PageRank perspective. *Inf. Process. Manag.* **2011**, *47*, 125–134. [CrossRef]
39. Li, J.; Xia, F.; Wang, W.; Chen, Z.; Asabere, N.Y.; Jiang, H. ACRec: A co-authorship based random walk model for academic collaboration recommendation. In Proceedings of the 23rd International Conference on World Wide Web, Seoul, Republic of Korea, 7–11 April 2014; pp. 1209–1214.
40. Balog, K.; Azzopardi, L.; Rijke, M.D. Formal models for expert finding in enterprise corpora. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, WA, USA, 6–11 August 2006; pp. 43–50.
41. de Campos, L.M.; Fernández-Luna, J.M.; Huete, J.F. A lazy approach for filtering parliamentary documents. In *Electronic Government and the Information Systems Perspective*; Lecture Notes in Computer Science; Kö, A., Francesconi, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9265, pp. 364–378.
42. de Campos, L.M.; Fernández-Luna, J.M.; Huete, J.F. Profile-based recommendation: A case study in a parliamentary context. *J. Inf. Sci.* **2017**, *43*, 665–682. [CrossRef]
43. Mimno, D.; McCallum, A. Expertise modeling for matching papers with reviewers. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), San Jose, CA, USA, 12–15 August 2007; pp. 500–509.
44. Balog, K.; Azzopardi, L.; Rijke, M.D. A language modeling framework for expert finding. *Inf. Process. Manag.* **2009**, *45*, 1–19. [CrossRef]
45. Liu, X.; Croft, W.B.; Koll, M. Finding experts in community-based question-answering services. In Proceedings of the 14th ACM International Conference on Information and Knowledge Management, Bremen, Germany, 31 October–5 November 2005; pp. 315–316.
46. Boeva, V.; Boneva, L.; Tsiporkova, E. Semantic-aware expert partitioning. In *Artificial Intelligence: Methodology, Systems, and Applications, AIMSA 2014*; Lecture Notes in Computer Science; Agre, G., Hitzler, P., Krisnadhi, A.A., Kuznetsov, S.O., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8722, pp. 13–24.
47. Li, C.; Cheung, W.K.; Ye, Y.; Zhang, X.; Chu, D.; Li, X. The Author-Topic-Community model for author interest profiling and community discovery. *Knowl. Inf. Syst.* **2015**, *44*, 359–383. [CrossRef]
48. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent Dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
49. Rosen-Zvi, M.; Griffiths, T.; Steyvers, M.; Smyth, P. The author-topic model for authors and documents. In Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, Banff, AL, Canada, 7–11 July 2004; pp. 487–494.
50. Tang, J.; Jin, R.; Zhang, J. A topic modeling approach and its integration into the random walk framework for academic search. In Proceedings of the 2008 IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 1055–1060.

51. Momtazi, S.; Naumann, F. Topic modeling for expert finding using latent Dirichlet allocation. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2013**, *3*, 346–353. [CrossRef]
52. Yang, C.; Ma, J.; Liu, X.; Sun, J.; Silva, T.; Hua, Z. A weighted topic model enhanced approach for complementary collaborator recommendation. In Proceedings of the 18th Pacific Asia Conference on Information Systems, Chengdu, China, 24–28 June 2014.
53. Pal, A.; Chang, S.; Konstan, J.A. Evolution of experts in question answering communities. In Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media, Dublin, Ireland, 4–7 June 2012; Volume 6.
54. Li, B.; King, I. Routing questions to appropriate answerers in community question answering services. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2010; pp. 1585–1588.
55. Zhou, T.C.; Lyu, M.R.; King, I. A classification-based approach to question routing in community question answering. In Proceedings of the 21st International Conference on World Wide Web, Lyon, France, 16–20 April 2012; pp. 783–790.
56. Zhang, D.; Zhao, S.; Duan, Z.; Chen, J.; Zhang, Y. A multi-label classification method using a hierarchical and transparent representation for paper-reviewer recommendation. *ACM Trans. Inf. Syst.* **2020**, *38*, 5. [CrossRef]
57. Wang, J.; Sun, J.; Lin, H.; Dong, H.; Zhang, S. Convolutional neural networks for expert recommendation in community question answering. *Sci. China Inf. Sci.* **2017**, *60*, 110102. [CrossRef]
58. Dehghan, M.; Rahmani, H.A.; Abin, A.A.; Vu, V. Mining shape of expertise: A novel approach based on convolutional neural network. *Inf. Process. Manag.* **2020**, *57*, 102239. [CrossRef]
59. He, T.; Guo, C.; Chu, Y. Enhanced user interest and expertise modeling for expert recommendation. In Proceedings of the 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 556–562.
60. Nikzad-Khasmakhi, N.; Balafar, M.; Feizi-Derakhshi, M.R.; Motamed, C. BERTERS: Multimodal representation learning for expert recommendation system with transformers and graph embeddings. *Chaos Solitons Fractals* **2021**, *151*, 111260. [CrossRef]
61. Sohangir, S.; Wang, D. Finding expert authors in financial forum using deep learning methods. In Proceedings of the Second IEEE International Conference on Robotic Computing, Laguna Hills, CA, USA, 31 January–2 February 2018; pp. 399–402.
62. de Campos, L.M.; Fernández-Luna, J.M.; Huete, J.F. Publication venue recommendation using profiles based on clustering. *IEEE Access* **2022**, *10*, 106886–106896. [CrossRef]
63. Zhang, M.L.; Li, Y.K.; Liu, X.Y.; Geng, X. Binary relevance for multi-label learning: An overview. *Front. Comput. Sci.* **2018**, *12*, 191–202. [CrossRef]
64. Tsoumakas, G.; Katakis, I. Multi-label classification: An overview. *Int. J. Data Warehous. Min.* **2007**, *3*, 1–13. [CrossRef]
65. Zhou, Z. *Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2021. [CrossRef]
66. Aggarwal, C.C. *Neural Networks and Deep Learning: A Textbook*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2023. [CrossRef]
67. Bauersfeld, L.; Romero, A.; Muglikar, M.; Scaramuzza, D. Cracking double-blind review: Authorship attribution with deep learning. *PLoS ONE* **2023**, *18*, e0287611. [CrossRef]
68. Tyo, J.; Dhingra, B.; Lipton, Z.C. On the state of the art in authorship attribution and authorship verification. *arXiv* **2022**, arXiv:2209.06869.
69. Prasad, R.S.; Chakkaravarthy, M. State of the art in authorship attribution with impact analysis of stylometric features on style breach prediction. *J. Cases Inf. Technol.* **2022**, *24*, 1–12. [CrossRef]
70. Prabhu, Y.; Varma, M. FastXML: A Fast, accurate and stable tree-classifier for eXtreme Multi-label Learning. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 263–272.
71. Kaufman, L.; Rousseeuw, P.J. *Finding Groups in Data: An Introduction to Cluster Analysis*; John Wiley: Hoboken, NJ, USA, 2005.
72. Can, F.; Ozkarahan, E. Concepts and effectiveness of the cover-coefficient-based clustering methodology for text databases. *ACM Trans. Database Syst.* **1990**, *15*, 483–517. [CrossRef]
73. Macdonald, C.; Ounis, I. Voting techniques for expert search. *Knowl. Inf. Syst.* **2008**, *16*, 259–280. [CrossRef]
74. de Campos, L.M.; Fernández-Luna, J.M.; Huete, J.F. Committee-based profiles for politician finding. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **2017**, *25*, 21–36. [CrossRef]
75. Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed.; O'Reilly Media, Inc.: Sebastopol, CA, USA , 2022.
76. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
77. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *arXiv* **2019**, arXiv:1908.10084.
78. de Campos, L.M.; Fernández-Luna, J.M.; Huete, J.F. An explainable content-based approach for recommender systems: A case study in journal recommendation for paper submission, submitted to User Model. *User Adapt. Interact.* 2024. *submitted*.
79. Neshati, M.; Hashemi, S.H.; Beigy, H. Expertise finding in bibliographic network: Topic dominance learning approach. *IEEE Trans. Cybern.* **2014**, *44*, 2646–2657. [CrossRef]
80. de Campos, L.M.; Fernández-Luna, J.M.; Huete, J.F. Use of topical and temporal profiles and their hybridisation for content-based recommendation. *User Model. User Adapt. Interact.* **2023**, *33*, 911–937. [CrossRef]