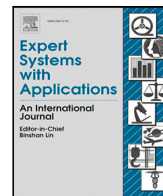





Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswaExpFinder: A hybrid model for expert finding from text-based expertise data Yong-Bin Kang^{a,*}, Hung Du^b, Abdur Rahim Mohammad Forkan^b, Prem Prakash Jayaraman^b, Amir Aryani^c, Timos Sellis^{d,1}^a ARC Centre of Excellence for Automated Decision-Making and Society, Swinburne University of Technology, Australia^b School of Science, Computing and Engineering Technologies, Swinburne University of Technology, Australia^c Social Innovation Research Institute, Swinburne University of Technology, Australia^d Archimedes Research Unit, Athena Research Center, Greece

ARTICLE INFO

Keywords:

ExpFinder
Expert finding
Expert collaboration graph
Bipartite graph
Text-based expertise
Co-hits

ABSTRACT

Finding an expert plays a crucial role in driving successful collaborations and speeding up high-quality research development and innovations. However, the rapid growth of scientific publications and digital data makes identifying the right experts a challenging problem. Existing approaches for finding experts given a topic can be categorised into information retrieval techniques such as vector space models, document language models, and graph-based models. In this paper, we propose *ExpFinder*, a new hybrid model for expert finding, that integrates a novel N -gram vector space model, denoted as $nVSM$, and a graph-based model, denoted as $\mu CO-HITS$, that is a proposed variation of the CO-HITS algorithm. The key of $nVSM$ is to exploit recent inverse document frequency weighting method for N -gram words, and *ExpFinder* incorporates $nVSM$ into $\mu CO-HITS$ to achieve expert finding. We comprehensively evaluate *ExpFinder* on four different datasets from the academic domains in comparison with six different expert finding models. The evaluation results show that *ExpFinder* is an highly effective model for expert finding, substantially outperforming all the compared models in 19% to 160.2%.

1. Introduction

Finding experts in a particular domain is key to accelerate rapid formation of teams to respond to new opportunities, as well as undertake and address new frontiers in research innovations. Further, accurately identified experts can significantly contribute to enhancing the research capabilities of an organisation leading to higher quality research outcomes (Han et al., 2019). In general, an *expert* is defined as a person who has sufficient 'knowledge and skills', called *expertise*, in a given field (Husain et al., 2019). While digitally available data describing experts' expertise is rapidly growing, manually collating such information to find experts seems impractical and expensive. Thus, often in a large research organisation with diverse disciplines, finding experts in a field that one does not know or has limited knowledge is particularly very challenging.

Information retrieval techniques have been widely used to aid retrieval task for finding experts from digitally available expertise data

(we collectively denote such data as *documents* in this paper) such as scientific publications (Stankovic et al., 2010). Based on the literature (Gonçalves & Dorneles, 2019), there are two specific tasks for expert retrieval: (1) *expert finding* - identifying experts given a topic from available documents and rank them based on their expertise level, and (2) *expert profiling* - identifying the areas of expertise given an expert. In this paper we focus on the expert finding task and propose a hybrid model for it from unstructured documents. We use the term *topic* to represent a field of expertise. Most existing approaches for expert finding can be classified into *vector space models* (VSM), *document language models* (DLM), and *graph-based models* (GM). In VSM, expert finding is often solved by modelling the weights of topics, associated with the documents produced by experts, using Term Frequency-Inverse Document Frequency (TFIDF) or its variation (Alhabashneh et al., 2017; Chuang et al., 2014). In DLM, expert finding is achieved by estimating the probability that a topic would be observed

The code (and data) in this article has been certified as Reproducible by Code Ocean: (<https://codeocean.com/>). More information on the Reproducibility Badge Initiative is available at <https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals>.

* Corresponding author.

E-mail addresses: ykang@swin.edu.au (Y.-B. Kang), hungdu@swin.edu.au (H. Du), fforkan@swin.edu.au (A.R.M. Forkan), pjayaraman@swin.edu.au (P.P. Jayaraman), aaryani@swin.edu.au (A. Aryani), timos@athenarc.gr (T. Sellis).

¹ Work done while at Swinburne University of Technology.

<https://doi.org/10.1016/j.eswa.2022.118691>

Received 4 September 2021; Received in revised form 23 August 2022; Accepted 23 August 2022

Available online 29 August 2022

0957-4174/© 2022 Elsevier Ltd. All rights reserved.

in the documents of an expert (Balog et al., 2009; Cifariello et al., 2019; Wang et al., 2015). In GM, a graph is often used to represent associations among experts, documents and/or topics. The strengths of the associations are inferred to estimate the expertise degree of an expert given a topic using various graph analytic methods such as expert–document–term association paths (Gollapalli et al., 2013), Hyper-Induced Topic Search (HITS) (Campbell et al., 2003; Yeniterzi & Callan, 2014), or social network analysis methods (Bok et al., 2019; Faisal et al., 2019; Sziklai, 2018).

In this work, we propose a hybrid model for expert finding, *ExpFinder*,² that integrates a novel N -gram VSM, denoted as n VSM, with a GM using an *expert collaboration graph* (ECG). We develop n VSM for estimating the expertise degree (or *weight*) of an expert given a topic by leveraging the recent IDF weighting (Shirakawa et al., 2017) for N -gram words (simply N -grams) composed of two or more terms (for $N > 1$). This method demonstrated a higher robustness and effectiveness in measuring the IDF weights of N -grams. We also build an ECG that is an expert–document bipartite graph to represent associations between experts and documents based on the co-authorship information. To estimate the weight of an expert given a topic on ECG, we propose the GM, μ CO-HITS, formed by applying two variation schemes to the generalised CO-HITS (Deng et al., 2009) algorithm.

Our motivation for developing *ExpFinder* lies in three reasons. First, despite promising performance of the IDF weighting (Shirakawa et al., 2017) for N -gram words, its nature and impact for expert finding has not been studied. This motivates us to investigate the design of n VSM that incorporates the IDF weighting for expert finding. Second, although n VSM utilises the advantageous features of the IDF weighting, it ignores the social importance (or influence) of experts. As demonstrated by Hyperlink-Induced Topic Search (HITS) (Campbell et al., 2003; Jiang et al., 2016; Yeniterzi & Callan, 2014) and PageRank (Koumenides & Shadbolt, 2014), utilisation of the link structure of entities under consideration in a network is useful for estimating the importance of the entities. To address the limitation of n VSM, we build an expert collaboration graph (ECG) to estimate such social importance, and design μ CO-HITS that is a variation form of CO-HITS. We have chosen CO-HITS as a baseline of μ CO-HITS. Although CO-HITS has been popularly used a fundamental algorithm that has been proven effective for ranking entities (e.g., web pages) on a bipartite graph in various applications (London & Csendes, 2013; Ma et al., 2022; Truong et al., 2022; Yang et al., 2020), utilising CO-HITS for expert finding has received little attention. Third, in order to exploit knowledge facets of both n VSM and μ CO-HITS, we develop a hybrid model, *ExpFinder*, that combines them. *ExpFinder* utilises and leverages both the graph (i.e., ECG) and content information (drawn from n VSM) from both sides (i.e., experts and documents), so as to improve the precision of expert finding.

This paper makes three main contributions. First, we present two novel expert finding models, n VSM and μ CO-HITS, where n VSM is a new VSM model utilising the IDF weighting, and μ CO-HITS explores the social importance of experts based on their indirect interactions via co-documents. Second, we present *ExpFinder* that is a hybrid model for expert finding to create a stronger expert finding model by combining n VSM and μ CO-HITS. We also provide in-depth explanation on how *ExpFinder* is built on, and show the *ExpFinder* achieves better performance than a single one (i.e., n VSM or μ CO-HITS alone). *ExpFinder* incorporates the weights of experts, estimated by n VSM, into an ECG, and uses μ CO-HITS to better estimate the weights of experts for a given topic. Third, we present a comprehensive evaluation that measures the outperformance of *ExpFinder* using four different datasets (LExR (Mangaravite et al., 2016) and three DBLP datasets (Bordea et al., 2013))

² *ExpFinder*'s source code is publicly available on <https://doi.org/10.24433/CO.3117588.v1>.

in academic domains, in comparison with six different expert finding models.

This rest of the paper is organised as follows. Section 2 provides related works in expert finding. Section 3 presents an overview of *ExpFinder* and Section 4 discusses in-depth steps for building *ExpFinder*. Section 5 presents thorough empirical evaluations of *ExpFinder*, followed by conclusion in Section 6.

2. Related work

In recent years, with the growing amount of digital expertise sources, expert finding has become an intensive research area in information retrieval community (Gonçalves & Dorneles, 2019). We can mainly classify expert finding approaches into three categories: VSM, DLM and GM.

In the VSM approach, the common idea is to estimate relevance between a document and a topic using a weighting scheme in VSM (e.g. TFIDF or its variation). Then, finding experts can be done by assuming that an expert is seen as the collection of its published documents D_x . That is, the weight of an expert x given a topic t is estimated by aggregating relevance scores between each document in D_x and t . For example, TFIDF was used to find experts in community question answering websites in which the goal is to find users with relevant expertise to provide answers for given questions (Riahi et al., 2012). A variation of TFIDF was also applied for expert finding in an organisation's ERP system (Schunk & Cong, 2010). The work (Chuang et al., 2014) also used TFIDF to identify experts given a topic using a topic extension method (finding interrelated terms of a given topic from the corpus), where TFIDF was used to estimate relevance between extended terms and each expert's documents. TFIDF was also used to estimate the weights of topics indicating the interests of an expert, and this information is used with fuzzy logics for expert finding (Alhabashneh et al., 2017).

The aim of the DLM approach is to find experts whose documents are directly related to a given topic. In common, this approach estimates the relationships between a topic and an expert as the probability of generating the topic by the expert (Balog et al., 2009), or between an expert and its publications (Mangaravite & Santos, 2016). *BMExpert* (Wang et al., 2015) used the DLM (Balog et al., 2009) for expert finding using three factors: relevance of documents to the topic, importance of documents, and associations between documents and experts. Similarly, the work (Van Gysel et al., 2016) used a probabilistic DLM for expert finding by probabilistically generating a textual representation of an expert according to his documents and then ranking such documents according to a given topic. Recently, a probabilistic model, *WISER* (Cifariello et al., 2019), estimated the importance of experts' documents given a topic using BM25 (Robertson & Zaragoza, 2009). Using this importance, such documents were ranked and these ranks were summed to represent the topic-sensitive weight of an expert.

In the GM approach, experts are represented as nodes, and their relationships are represented by their edges or implicitly derived from a graph. Different algorithms were used in the GM approach, such as Hyperlink-Induced Topic Search (HITS) (Campbell et al., 2003; Jiang et al., 2016; Yeniterzi & Callan, 2014) and PageRank (Koumenides & Shadbolt, 2014). For expert finding, PageRank was adapted in the context of online community discussions on a user–user graph built based on votes from users whose questions were answered by whom (Zhang et al., 2007). Also, a modified PageRank algorithm was developed and applied for finding experts in online knowledge communities (Wang et al., 2013). HITS is also a graph-based link analysis algorithm originally designed for ranking the importance of web pages based on authority and hub scores. The work (Campbell et al., 2003) built an expert–expert bipartite graph based on email communication patterns and attempted to find the ranking of experts using HITS. CO-HITS was introduced (Deng et al., 2009) to incorporate a bipartite graph

with the content information from both sides (e.g. experts and documents in our context) by adding personalised parameters to HITS, and CO-HITS showed higher performance than HITS (Deng et al., 2009). Using an author–document–topic (ADT) graph, the expert finding GM model (Gollapalli et al., 2013) leveraged possible paths between a topic and an expert on the ADT graph. Recently, diverse expert finding approaches were proposed in a social network. For example, the authors (Faisal et al., 2019) proposed a method for finding experts who can answer questions in a social network for ‘community question answering’ using users’ votes and reputations. The approach (Bok et al., 2019) focused on finding experts who can answer users’ questions based on users’ online social activities in a social network (e.g. Twitter).

Also, we observe that some models tend to mix different techniques from DLM, VSM, and/or GM. For example, AuthorRank (Deng et al., 2011) combined a generative probabilistic DLM and a co-authorship network based on the awareness of community information of expert candidates. The DLM was used to identify the most relevant documents, while the network was used to model the authors’ authorities based on the community co-authorship. The work (Liu et al., 2013) combined a cluster-based language model and a VSM for finding experts in question and answer communities. The authors (Kundu & Mandal, 2019) proposed a complex model for community question answering using a variation of the DLM (Balog et al., 2009) and a HITS-based GM (the HITS algorithm on a competition based expertise network (Aslay et al., 2013)), where the scores from these models were linearly combined to rank experts given a question. The work (Torkzadeh Mahani et al., 2018) used the Dempster-Shafer combination theory to combine the DLM (Balog et al., 2009) and a graph algorithm that analyses a social interaction of experts.

Although VSM and DLM approaches exploit the content information of expertise of experts, one common problem is that they likely ignore the social influence of experts who may have similar expertise. Utilising such social influence information has been shown to be another valuable information source for identifying experts as highlighted in GM approaches. GM approaches focus more on interactions between experts in expert social networks, thus the content information of expertise tends to be less utilised. Instead, ExpFinder is a hybrid model that incorporates the knowledge facets of both VSM and GM. ExpFinder is a combination form of our novel VSM model (n VSM) and GM model (μ CO-HITS). n VSM takes advantage of the IDF weighting for N -grams (Shirakawa et al., 2017), and ExpFinder feeds n VSM into μ CO-HITS that is a novel variation of CO-HITS to improve precision of expert finding. Moreover, aforementioned mixture models were mostly applied in finding experts in question-answering communities on social networks. Differing from them, we focus on the task of experts finding in academia, aiming to find experts whose expertise is represented through their text-based evidence (e.g., publications) given a topic.

3. Introduction to ExpFinder

In this section, we present the overview of ExpFinder, and the basic notations that we will use in the paper.

3.1. Overview of ExpFinder

ExpFinder aims to identify ranked experts according to their expertise degree given a topic. In this paper, we assume that a topic is represented as a *noun phrase* which is extracted from documents (e.g. scientific publications) of experts in a given domain. The reason is that domain-specific concepts are often described by noun phrases that represent the key information within a given corpus (Kang et al., 2014). A noun phrase means a single-word noun or a group of words that function together as a noun.

The key of ExpFinder is the utilisation of *two knowledge facets* in a unified manner. The one is the estimation of the weights of experts given a topic by utilising information in the proposed n VSM. The

second facet is μ CO-HITS that performs on an expert collaboration graph (ECG), where the expert collaboration is measured by the joint production of experts (e.g. co-authored documents). We incorporate the result of n VSM into the ECG, and reinforce the weights of experts given a topic using μ CO-HITS. The following presents the key steps in ExpFinder (see also Fig. 1):

Step 1: Extract topics: Given experts and their documents (also called *corpus*) in a given domain, we extract noun phrases as topics.

Step 2: Estimate the weights of experts and documents given topics: Given a topic, we estimate the weights of experts and documents based on the proposed n TFIDF method in n VSM. In this paper, we also call such weights *topic-sensitive weights* as these weights are sensitive to the given topic. Given a topic, the key of n TFIDF lies in a combination of the frequency of the topic with the IDF method of N -grams over the corpus (Shirakawa et al., 2017). The output of this step includes a topic–expert matrix and a topic–document matrix, where an entry reflects the weight of an expert and a document given a topic, respectively.

Step 3: Construct an ECG: We construct an ECG to represent associations between experts and their jointly-published documents. This graph is modelled by a *directed, weighted bipartite graph* that has two kinds of nodes, one representing experts and the other representing documents. A directed edge points from a document d to an expert x , if x has published d .

Step 4: Reinforce expert weights using μ CO-HITS: As presented above, to rank experts, ExpFinder integrates the two knowledge facets: (1) n VSM to estimate the weights of the experts and documents given a topic (Step 2); and (2) μ CO-HITS incorporating such weights into an ECG (Step 3) to further reinforce the weights of experts. The outcome of this step is the reinforced topic–expert matrix showing the weights of experts. Finally, we rank the experts for each topic from the matrix.

3.2. Notations

We present the following basic notations in this paper.

- Let \mathcal{X} be the set of experts, and $|\mathcal{X}|$ be the number of experts in \mathcal{X} .
- Let \mathcal{D} be the set of all documents published by \mathcal{X} . Let \mathcal{D}_x be all documents published by $x \in \mathcal{X}$. Also, let \mathcal{X}_d denotes the set of the experts that have a document $d \in \mathcal{D}$.
- Let \mathcal{T} be the set of topics extracted from \mathcal{D} .
- Let \mathbf{TX} be a $|\mathcal{T}| \times |\mathcal{X}|$ topic–expert matrix where rows and columns are labelled with \mathcal{T} and \mathcal{X} , respectively. The entry that lies in the i th row and the j th column of \mathbf{TX} is denoted as $\mathbf{TX}_{i,j}$ that indicates the weight of $x_j \in \mathcal{X}$ on $t_i \in \mathcal{T}$. If a weight is higher, the more important the corresponding expert is on the given topic.
- Let \mathbf{DX} be a $|\mathcal{D}| \times |\mathcal{X}|$ document–expert matrix where rows and columns are labelled with \mathcal{D} and \mathcal{X} , respectively. The entry of $\mathbf{DX}_{i,j}$ shows the weight of an expert $x_j \in \mathcal{X}$ on a document $d_i \in \mathcal{D}$ based on x_j ’s contribution towards d_i .
- Let \mathbf{TD} be a $|\mathcal{T}| \times |\mathcal{D}|$ topic–document matrix where rows and columns are labelled with \mathcal{T} and \mathcal{D} , respectively. $\mathbf{TD}_{i,j}$ represents the weight of document $d_j \in \mathcal{D}$ on $t_i \in \mathcal{T}$. If a weight is higher, the more important the corresponding document is on the given topic.

4. Design of ExpFinder

In this section, we present the details of the four steps for designing and developing ExpFinder.

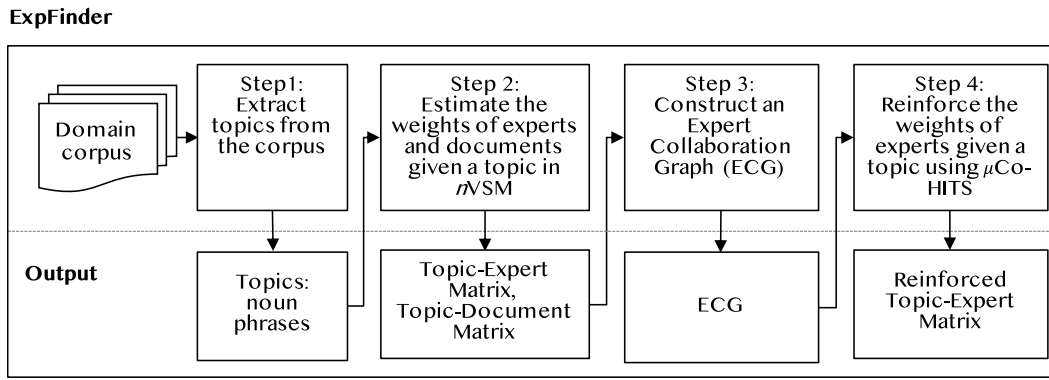


Fig. 1. The overview of ExpFinder.

4.1. Extract topics

As presented in Section 3, we assume that a topic is represented as a noun phrase. We perform the following steps to extract noun phrases from D . First, for each document $d \in D$, we split d into its sentences keeping their sequential indices. Second, for each sentence, we analyse POS tags of the words in the sentence and remove stopwords. POS tagging is the process for assigning a part of speech to each word in a sentence. Then, each word remained is converted into its lemmatised form. Lemmatisation is the process of grouping together the inflected forms of a word, thus they can be considered to be a single item (e.g. ‘patients’ is lemmatised to ‘patient’). Third, in the sentence, we use the following linguistic pattern based on POS tags to extract noun phrases:

$$(JJ)^*|(VBN)^*|(VBG)^*(N)^+, \quad (1)$$

where ‘JJ’ means adjective, ‘VBN’ past participle, ‘VBG’ gerund, and ‘N’ nouns. Using this pattern, we can extract a noun phrase starting with (1) one or more nouns; (2) one or more adjectives followed by one or more nouns (e.g. ‘medical system’); (3) one or more past participle followed by one or more nouns (e.g. ‘embedded system’); and (4) one or more gerund followed by one or more nouns (e.g. ‘learning system’). The symbol ‘*’ denotes zero or more occurrences, ‘+’ denotes one or more occurrences.

Note that ExpFinder does not rely on a particular method for noun phrase extraction, and thus can incorporate any noun phrase extraction methods.

4.2. Estimate the weights of experts and documents given topics

We now present the process for creating a topic–expert matrix \mathbf{TX} and a topic–document matrix \mathbf{TD} from the extracted topics using n TFIDF in n VSM. These matrices will be used as the input to μ CO-HITS.

4.2.1. Topic–expert matrix creation

To create a \mathbf{TX} , our fundamental is to utilise the definition of the DLM (Balog et al., 2009; Wang et al., 2015) for expert finding. Thus, we first briefly describe how this DLM can measure the topic-sensitive weight of an expert $x \in \mathcal{X}$ given a topic $t \in \mathcal{T}$, denoted as $p(x|t)$. Formally, it is given as (Balog et al., 2009):

$$p(x|t) = p(x, t)/p(t), \quad (2)$$

where $p(x, t)$ is the joint probability of x and t , and $p(t)$ is the probability of t . We ignore $p(t)$ as this is a consistently constant over all experts \mathcal{X} . Thus, $p(x|t)$ is approximated by $p(x, t)$ that is reformulated considering documents D_x (Balog et al., 2009):

$$\begin{aligned} p(x, t) &= \sum_{d \in D_x} p(x, d, t) = \sum_{d \in D_x} p(d)p(x, t|d) \\ &= \sum_{d \in D_x} p(d)p(t|d)p(x|d). \end{aligned} \quad (3)$$

In Eq. (3), we observe the following notations (Wang et al., 2015):

- $p(d)$ is the prior probability of d that can also be interpreted as the weight (or importance) of d .
- $p(x|d)$ is the conditional probability of x given d (e.g. in a simply way, it can be estimated based on the order of x in the co-author list in d (Wang et al., 2015)).
- $p(t|d)$ is the conditional probability of t given d .

In the DLM (Balog et al., 2009; Wang et al., 2015), it is assumed that a document d is described as a collection of terms that appear in d . An importance of a term $w \in t$ within d is determined by the proportion of its occurrences. DLMs provide a way of capturing this notion by representing a document as multinomial probability distribution over the vocabulary of terms. To estimate $p(t|d)$, let θ_d be the document model of d , and the probability of t in θ_d is $p(t|\theta_d)$. This $p(t|\theta_d)$ indicates how likely we see t if we sampled t randomly from d . Thus, $p(t|d)$ is rewritten as $p(t|\theta_d)$ taking the product of t ’s individual term probabilities as follows (Balog et al., 2009; Wang et al., 2015):

$$p(t|\theta_d) = \prod_{w \in t} p(w|\theta_d), \quad (4)$$

where w is an individual term in t . However, a limitation of Eq. (4) is that unseen terms in d would get a zero probability. Thus, it is a common in DLMs to introduce a *smoothing* factor to assign non-zero probability to the unseen terms. Typically, it can be done by reducing the probabilities of the terms seen in the corpus and assigning the additional probability mass to unseen terms. Formally, $p(w|\theta_d)$ is re-expressed as:

$$p(w|\theta_d) = (1 - \lambda_\theta)p(w|d) + \lambda_\theta p(w|D) \quad (5)$$

where $p(w|d)$ is estimated by the term frequency of w in d divided by $|d|$ (the number of terms in d), denoted as $tf(w, d)$, and $p(w|D)$ is the term frequency of w in D normalised by $|D|$, i.e., $tf(w, |D|)$. The parameter λ_θ controls the influence of the two probabilities.

We now present our novelty for estimating $p(t|d)$ using n TFIDF. Since n TFIDF is an extension of TFIDF, we briefly describe how $p(t|d)$ can be estimated using TFIDF in VSM. In a sense, $p(t|d)$ can also be interpreted using TFIDF (Roelleke & Wang, 2008). Note that TFIDF is a measure based on the distance between two probability distributions, expressed as the cross-entropy: (1) a *local distribution* of $w \in t$ in d , and (2) a *global distribution* of w in D . TFIDF is a measure of perplexity between these two distributions. A higher perplexity score implies a higher relevance of d to w . The cross-entropy between distributions p_w and q_w is as follows:

$$-\sum_w p_w \log q_w = \sum_w p_w \log \frac{1}{q_w}, \quad (6)$$

if we substitute p_w with $tf(w, d)$ (TF) and $\frac{1}{q_i}$ with the inverted probability of encountering d with a term w (IDF), denoted as $\frac{|D|}{df(w)}$, where

$df(w)$ is the document frequency of w , we obtain a TFIDF formula:

$$p(t|d) \approx \sum_{w \in t} tf(w, d) \log \frac{|D|}{df(w)}. \quad (7)$$

Thus, as highlighted in Lu (2013), VSM and DLM are actually closely related. The TF component $tf(w, d)$ is exactly same as the probability of seeing a term w in DLM. The IDF component $\frac{|D|}{df(w)}$ is implicitly related to a smoothing method in DLM that uses the collection frequency ($tf(w, |D|)$): term frequency of w in D normalised by $|D|$.

Based on the above observation, we now present our approach for estimating $p(t|d)$ using n TFIDF in n VSM. Although some variant forms of TFIDF methods have been proposed, the majority of TFIDF methods use the same IDF function (Shirakawa et al., 2017). However, one drawback of IDF is that it cannot handle N -grams, contiguous sequence of N terms (for $N > 1$). The reason is that IDF tends to give a higher weight to a term that occurs in fewer documents. Note that typically, phrases occur in fewer documents when their collocations are less common. Thus, uncommon phrases (e.g. noise phrases) are unintentionally assigned high weight, yielding the conflict with the definition of a good phrase that constitutes a succinct conceptual descriptor in text. To address it, N -gram IDF for weighting phrases was recently proposed (Shirakawa et al., 2017). N -gram IDF has shown the ability to accurately estimate weights of dominant phrases of any length, simply using the domain corpus.

The key in n VSM is the proposed formula n TFIDF that uses a combination of the frequency of a topic t with t 's N -gram IDF. As that frequency, we use the average frequencies of the constituent terms in t . Formally, using n TFIDF, $p(t|d)$ is defined as:

$$p(t|d) \approx nTFIDF(t, d) = nt f(t, d) \cdot nid f(t), \text{ where}$$

$$nt f(t, d) = \frac{\sum_{i=1}^n tf(w_i, d)}{|t|}, \quad (8)$$

$$nid f(t) = \log \frac{|D| \cdot df(t)}{df(w_1 \wedge w_2 \wedge \dots \wedge w_n)^2}$$

where w_1, \dots, w_n are n -constituent terms in t , $tf(w_i, d)$ is the term frequency of w_i in d normalised by $|d|$, $|t|$ is equal to n , and $nidf(t)$ is the N -gram IDF method for t (Shirakawa et al., 2017). Eq. (8) applies for all $|t| \geq 1$, where $nidf(t)$ is equal to the log-IDF, $\log \frac{|D|}{df(t)}$, in Eq. (7), when $|t|=1$. Finally, in n VSM, $p(x|t)$ in Eq. (3) is calculated using $p(t|d)$ in Eq. (8) and stored into $\mathbf{TX}_{i(t),i(x)}$, where $i(t)$ and $i(x)$ indicate the row and column index of t and x , respectively, in the \mathbf{TX} .

4.2.2. Topic–document matrix creation

To create a topic–document matrix \mathbf{TD} , we need to calculate the topic-sensitive weight of a document d given a topic t . Following the idea of the DLM (Balog et al., 2009) again, we estimate this weight by calculating the probability of d being relevant to t : $p(d|t)$. Using the Bayes theorem, $p(d|t)$ can be calculated as:

$$p(d|t) = p(t|d)p(d)/p(t), \quad (9)$$

where $p(t)$ can be ignored as it is a consistently constant over all documents. Thus, $p(t|d)p(d)$ can be calculated by multiplying $p(t|d)$ in Eq. (8) and $p(d)$. Finally, $p(d|t)$ is stored into the entry $\mathbf{TD}_{i(t),i(d)}$, where $i(t)$ and $i(d)$ indicate the row and column index of t and d , respectively, in the \mathbf{TD} .

4.3. Construct an ECG

Although we have estimated the topic-sensitive weight of an expert x for a given topic t in n VSM, one potential limit may be that $p(x|t)$ in Eq. (3) mainly relies on the documents D_x (i.e. $\sum_{d \in D_x}$), ignoring the social importance (or influence) of experts. Our premise is that the expertise degree of x on t can depend not only on x 's knowledge on t , but also on x 's social importance among t 's collaborating experts in a given domain. Thus, we propose that an expert collaboration graph (i.e. ECG) can also be a valuable source, in order to estimate

such social importance. This estimation is achieved by identifying more *authoritative* (or influential) topic-sensitive experts considering their joint documents. That is, in a sense, an ECG is a social network for experts, and ExpFinder calculates the authority score of x by repeatedly exploring the collective importance of the joint documents, published by x and x 's coauthors, using μ CO-HITS over the ECG. More specifically, ExpFinder incorporates the topic-sensitive weights of experts given topic, estimated by n VSM, into an ECG and reinforces such weights using μ CO-HITS.

Let $G = (V, E)$ be an ECG (i.e. directed, weighted bipartite graph) that has two node types: experts \mathcal{X} (also called *authorities*) and documents D (also called *hubs*). Thus, the node set $V = \mathcal{X} \cup D$. In G , each expert is not connected to any other experts, and the same is with the documents. A directed edge points from a document $d \in D$ to an expert $x \in \mathcal{X}$, if x has the authorship on d . This edge is denoted as e_{dx} . Thus, the set of edges E contain directed edges from D to \mathcal{X} . Given e_{dx} , its weight, denoted as w_{dx} , comes from $\mathbf{DX}_{i(d),i(x)}$ (see Section 3.2).

An example ECG is depicted in Fig. 2(a), where the solid lines show associations between experts and documents. The dashed arrows show implicit collaborations between experts via their joint documents: e.g., x_1 and x_2 have the joint documents d_1 and d_2 , such that a collaboration between them is established as a bidirectional dashed arrow.

4.4. Reinforcing expert weights using μ CO-HITS

Note that μ CO-HITS is a variation of CO-HITS (Deng et al., 2009). Thus, we first present the basic notion of CO-HITS on the structure of ECG. That is, an important document is expected to point to important experts, while an important expert is linked by important documents. The importance of an expert x is called the *authority score* of x , and the importance of a document d is called the *hub score* of d . These scores are non-negative weights. Here, our goal is to reinforce the topic-sensitive weights of experts, estimated by n VSM, using μ CO-HITS on the underlying ECG. For this, our idea is that given a topic t , we propagate the authority and hub scores with respect to t by traversing \mathcal{X} and D on the ECG via an iterative process.

An example is shown in Fig. 2(b), where the hub scores, $H(d_1)$ and $H(d_2)$, are propagated to the expert x_1 to update the authority score $A(x_1)$; $H(d_2)$ is also propagate to update $A(x_2)$; and $H(d_3)$ is propagated to update $A(x_2)$ and $A(x_3)$. Once all the authority scores are updated, these scores are again propagated to the hubs to update their scores. This performs iteratively. The intuition behind the iteration is the repeated mutual reinforcement to estimate authority and hub scores from co-linked nodes on the ECG.

In order for ExpFinder to incorporate an topic into μ CO-HITS, we take two steps. First, we extend the CO-HITS equation (Deng et al., 2009) to accommodate a topic. We call this extension *topic-sensitive* CO-HITS. As the initial authority and hub scores, our key idea is to use the estimated topic-sensitive weights of experts and documents in n VSM, respectively. Second, we newly design and apply our variation of topic-sensitive CO-HITS into the ECG. We elaborate these two steps in the rest of this section.

As the first step, we formally present the topic-sensitive CO-HITS equation, given an expert x and a topic t :

$$A(x; t)^k = (1 - \lambda_x) \alpha_{x,t} + \lambda_x \sum_{e_{dx} \in E} w_{dx} H(d; t)^{k-1}$$

$$H(d; t)^k = (1 - \lambda_d) \alpha_{d,t} + \lambda_d \sum_{e_{du} \in E} w_{du} A(u; t)^k \quad (10)$$

where

- $A(x; t)^k$ and $H(d; t)^k$ are the topic-sensitive authority score of x and topic-sensitive hub score of d , respectively, given t at k th iteration.

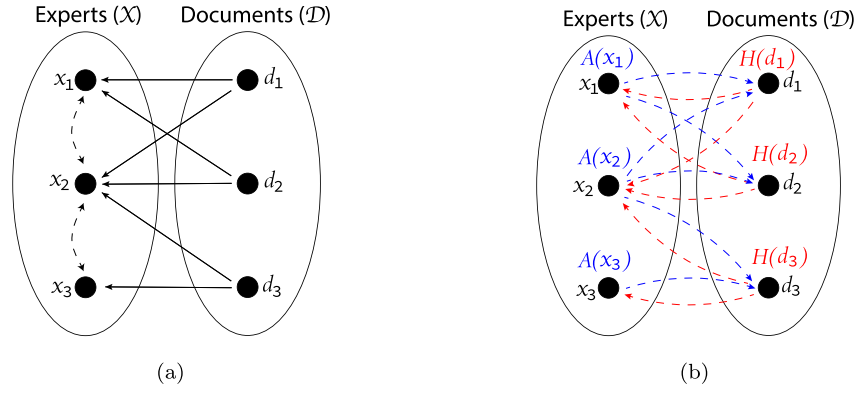


Fig. 2. An example ECG (a) and the score propagation of authority and hub nodes (b).

- w_{dx} denotes the weight of the edge e_{dx} , and thus $w_{dx} = \mathbf{DX}_{i(d),i(x)}$ and $w_{du} = \mathbf{DX}_{i(d),i(u)}$, where $i(d)$ and $i(x)$ indicate the row and column index of d and x , respectively, on \mathbf{DX} .
- k indicates a iteration number starting from 1.
- $\alpha_{x;t}$ is the initial score for $A(x;t)^*$ and $\alpha_{d;t}$ is the initial score for $H(d;t)^*$ given t . We call these scores *personalised weights*. In this work, these personalised weights are normalised to be the widely used L2-norm (Kleinberg, 1999), that is, $(\sum_{x_i \in \mathcal{X}} \alpha_{x_i;t})^{1/2} = 1$ and $(\sum_{d_i \in \mathcal{D}} \alpha_{d_i;t})^{1/2} = 1$. Also, after updating the k th iteration, the square root of the sum of squares of $A(x;t)^k$ and $H(d;t)^k$ are normalised using L2-norm, respectively. Assigning the personalised weights provides crucial information in CO-HITS as they provide valuable and make an impact on the propagation of the updates of both authority and hub scores (Deng et al., 2009). Our approach to determining the personalised weights is presented when discussing our proposed variation equation of Eq. (11).
- $\lambda_x \in [0, 1]$ and $\lambda_d \in [0, 1]$ are *personalised parameter* for expert and document, respectively. These parameters determine how much we consider the personalised weights when calculating the k th scores. Assigning lower values indicates that higher importance is given to the personalised weights while reducing the propagation effects of co-linked nodes.
- Using Eq. (10), the topic-sensitive CO-HITS algorithm performs as follows: (1) with the personalised weights, a user-specified k and a topic t , update all authority scores of \mathcal{X} ; and (2) update all hub scores of \mathcal{D} . These steps are repeatedly performed k times.

In the second step, we design the μ CO-HITS equation and apply it on the underlying ECG using two variation schemes of topic-sensitive CO-HITS:

$$\begin{aligned}
 A(x;t)^k &= (1 - \lambda_x)A(x;t)^{k-1} + \lambda_x \left(\frac{\sum_{e_{dx} \in E} w_{dx} H(d;t)^{k-1}}{\sum_{e_{dx} \in E} w_{dx}} \right) \\
 H(d;t)^k &= (1 - \lambda_d)H(d;t)^{k-1} + \lambda_d \left(\frac{\sum_{e_{du} \in E} w_{du} A(u;t)^k}{\sum_{e_{du} \in E} w_{du}} \right)
 \end{aligned} \tag{11}$$

where the interpretation of all the variables is the same as presented for Eq. (10), except the following two variation schemes.

The first variation scheme is that rather than using the *fixed* personalised weights $\alpha_{x;t}$ and $\alpha_{d;t}$, μ CO-HITS uses *dynamic* personalised weights $A(x;t)^{k-1}$ and $H(d;t)^{k-1}$ at each k th iteration. In Eq. (10), regardless of iterations, the authority and hub scores at each iteration are fixed to be $\alpha_{x;t}$ and $\alpha_{d;t}$. Different from it, our approach is to use personalised weights at the k th iteration as the $(k-1)$ -th authority and hub scores. By doing so, in the calculation of the authority (resp. hub) scores at the k th iteration, the our aim is to exploit both the propagation of the hub (resp. authority) scores and the effect of the authority (resp. hub) score at the $(k-1)$ -th iteration. Thus, in μ CO-HITS, personalised weights are updated at each iteration based on the authority and

hub scores at the previous iteration. In our approach, as the initial personalised weights, we use the topic-sensitive weights of experts and documents estimated using n TFIDF in n VSM. Thus, $A(x;t)^0 = \mathbf{TX}_{i(t),i(x)}$ and $H(d;t)^0 = \mathbf{TD}_{i(t),i(d)}$. Similarly, in the topic-sensitive CO-HITS equation in Eq. (10), $\alpha_{x;t}$ and $\alpha_{d;t}$ are set to be $A(x;t)^0$ and $H(d;t)^0$, respectively. By doing so, we integrate n VSM with μ CO-HITS, generating a new unified formula for this integration. Our intuition for this integration is to improve the accuracy for expert finding by further exploring the implicit relationships between experts, derived from the ECG, in addition to the results of the n VSM approach. Note that n VSM ignores such relationships, only utilising the importance of a document d ; the importance of a topic t from the documents of an expert x ; and the importance of x given d (see Eq. (3)).

The second variation scheme is that the *aggregation* of the authority and hub scores is different from that of topic-sensitive CO-HITS. In Eq. (10), $A(x;t)^k$ and $H(d;t)^k$ are calculated based on the square root of the *sum* of squares of $H(d;t)^{k-1}$ and $A(x;t)^k$, respectively. This approach tends to assign a higher authority score to an expert x who has *more* documents (i.e. $|\mathcal{D}_x|$). Similarly, it is likely that a higher hub score is given to a document d that is linked to *more* experts (i.e. $|\mathcal{X}_d|$) that have d .

Instead, in μ CO-HITS, we use the *central tendency* of $H(d;t)^{k-1}$ to calculate $A(x;t)^k$; and also use the *central tendency* of $A(x;t)^k$ to calculate $H(d;t)^k$. The ‘average’ is used to measure such central tendency. The reason is that we have already incorporated the idea of using ‘the sum of squares of authority and hub scores’, used in topic-sensitive CO-HITS, in the context of n VSM. Note that in n VSM, we calculated the topic-sensitive weights of experts by using the *sum* operator as seen in Eq. (3) (i.e. $\sum_{d \in \mathcal{D}_x}$). Thus, to avoid the duplicated use of this ‘sum’ operator, given a topic t , we design μ CO-HITS in a way that estimates the importance of an expert x at the k th iteration (i.e. $A(x;t)^k$) by calculating the *average* of the $(k-1)$ -th hub scores, in addition to personalised weight $A(x;t)^{k-1}$. Similarly, we estimate the importance of a document d (i.e. $H(d;t)^k$) by calculating the *average* of the k th authority scores, in addition to personalised weight $H(d;t)^{k-1}$. In the name μ CO-HITS, ‘ μ ’ indicates the ‘average’ so that μ CO-HITS means a particular topic-specific CO-HITS using the ‘average’ importance of authority and hub scores.

We also highlight other features of μ CO-HITS. First, as with topic-sensitive CO-HITS, the updated authority and hub scores at each iteration are normalised using L2-norm. Second, if λ_x and λ_d are 0, μ CO-HITS returns the initial personalised weights at each iteration. Thus, ExpFinder does not use the score propagation effects on the ECG, returning the same results obtained from n VSM. Third, if λ_x and λ_d are all equal to 1, μ CO-HITS does not incorporate personalised weights. However, it calculates $H(x;t)^1$ based on the $H(d;t)^0$ that was obtained from the topic–document matrix \mathbf{TD} , i.e., $H(d;t)^0 = \mathbf{TD}_{i(t),i(d)}$, generated by n VSM. Also, $H(d;t)^1$ is calculated based on $A(u;t)^1$.

Table 1
A summary of our four datasets.

	LExR	IR	SW	CL
# of documents	14 879	2355	1519	1667
# of experts	620	276	394	358
# of topics	227	268	2046	1583
Avg. # of documents per expert	28	9	4	5
Avg. # of experts per topic	6	10	9	8
Median # of experts per topic	5	8	6	5
Max # of experts per topic	26	177	226	158

5. Evaluation of ExpFinder

To assess the effectiveness of ExpFinder, we conduct the following evaluation. First, we measure the effectiveness of the first knowledge facet of ExpFinder, n VSM, in comparison with TFIDF-based VSM and two DLM approaches (Balog et al., 2009; Cifariello et al., 2019) (Section 5.3). Second, we show how to empirically find the best values for personalised parameters of ExpFinder (Section 5.4). Third, we evaluate that ExpFinder is a highly competitive model for expert finding, in comparison with n VSM and three hybrid approaches that combine certain forms of VSMs and GMs (Gollapalli et al., 2013; Schall, 2015) (Section 5.5). Finally, we summarise our evaluation results (Section 5.6).

5.1. Datasets

We use four benchmark datasets in our evaluation. One is the Lattes Expertise Retrieval (LExR)³ test collection (Mangaravite et al., 2016) for expertise retrieval in academic. LExR provides a comprehensive, large-scale benchmark for evaluating expertise retrieval and it covers all knowledge areas (e.g. earth sciences, biology, health sciences, languages, art, etc.) working in research institutions all over Brazil. Most publications are written in Portuguese, Spanish and English. In our evaluation, we only consider the English documents for our readability. The other three datasets⁴ are Information Retrieval (IR), Semantic Web (SW), and Computational Linguistics (CL) which are filtered subsets of DBLP dataset (Bordea et al., 2013). In these four datasets, we regard the authors as experts \mathcal{X} and the publications as documents \mathcal{D} , where each publication is seen as a mixture of title and abstract. From \mathcal{D} , we extract phrases as the first step in ExpFinder (Section 3).

These datasets also provide the ground-truth about who are the known experts for the known topics. The expert degrees for each topic are represented as non-relevance, relevance, and high relevance in LExR. We regard individuals with non-relevance as *non-experts*, and individuals with relevance and high relevance as *experts*. IR, SW and CL also provide the expert list for each topic. We formalise the candidates in such list as *experts*, and otherwise *non-experts*. From each dataset, we preprocess the following steps to be used in our evaluation. First, we remove publications containing empty title and abstract. Second, we remove publications whose abstracts provide little information, that is, less than 5 words after removing stopwords. Third, if there exists duplicated topics, we remove such ones.

Table 1 shows an overview of the datasets after performing these steps.

We note that our chosen datasets are relatively more comprehensive than some previous works, which focused on academic domains for their evaluation, in terms of the number of topics considered, thereby providing a reasonable measure of the effectiveness of ExpFinder. For example, the works (Deng et al., 2008), (Gollapalli et al., 2013)

and (Wang et al., 2015) used two datasets with seven topics, two datasets with 13 and 203 topics and one dataset with 14 topics, respectively.⁵ Note that our evaluation have been done using the larger numbers of the topics on the four datasets as seen in Table 1.

5.2. Evaluation framework

We present our evaluation configuration and metrics. Recall that as a topic, we use a phrase. We assume that the maximum word length of each phrase is 3 in our evaluation. Also, we observe that there is no guarantee that an original known topic t_g always appears in documents D in each dataset. Thus, given each t_g , we find its most similar phrase t from D . Then, t is alternatively used as a topic, instead of t_g . To find t given t_g , we use the scientific language model SciBERT⁶ (Beltagy et al., 2019) that is a scientific language model trained on the fulltext of 1.14M papers and 3.1B words, where the papers were collected from ‘semanticscholar.org’. Using this model allows us to measure a semantic similarity between t_g and t by their cosine similarity according to their corresponding vectors represented in the model. More specifically, assume that s_1 is an original known topic and s_2 is a phrase extracted from D . Then, we measure their similarity as:

$$\text{sim}(s_1, s_2) \approx \cos(\vec{s}_1, \vec{s}_2) = \frac{\vec{s}_1 \cdot \vec{s}_2}{\|\vec{s}_1\| \|\vec{s}_2\|}, \quad (12)$$

where \vec{s}_1 and \vec{s}_2 are the represented vectors of s_1 and s_2 in SciBERT, respectively. Each of these vectors is estimated by the average of the embedded vectors of its constituent terms. Suppose that s_1 consists of n -terms, $s_1 = (w_1, \dots, w_n)$, then, $\vec{s}_1 \approx \frac{1}{n}(\vec{w}_1 + \dots + \vec{w}_n)$, where $(\vec{w}_1, \dots, \vec{w}_n)$ are the embedded vectors of (w_1, \dots, w_n) . The same principle is applied to s_2 . Table 2 shows the examples of five topic-phrase pairs in each dataset, where each pair shows an original known topic t_g and the most similar phrase t used as a topic in our evaluation. As we see, some phrases (t) are equal to the original topic (t_g), while some others phrases are semantically very similar to the corresponding original topic (e.g. ‘image classification’-‘image recognition’ on CL).

Other evaluation configuration includes: (1) we assume that the importance of documents is the same (i.e. $p(d)=1$) and the importance of all experts of d is the same (i.e. $p(x|d)=1$) in Eq. (3). The reason is that one of our primary focuses is to evaluate the capability of n TFIDF in n VSM in calculating $p(t|d)$ in Eq. (3); (2) Thus, we also fix $w_{dx}=1$ and $w_{dt}=1$ in Eqs. (10) and (11); and (3) from our empirical testing, we observed Eqs. (10) and (11) are commonly converged after 5 iterations, so we set $k = 5$.

For all expert finding models in our evaluation, our aim is to generate a ranked list of experts given a topic. We use two widely-used evaluation metrics for expert finding (Deng et al., 2009; Wang et al., 2015): (1) *precision at rank n* ($P@n$) and (2) *Mean Average Precision* (MAP). $P@n$ measures the relevance of the n -top ranked experts with respect to a given query topic, defined as (Deng et al., 2009):

$$P@n = |S_n \cap R_t|/n, \quad (13)$$

where S_n is the set of n -top recommended experts for a given topic t , and R_t is the set of known experts for t . We report from $P@10$ to $P@30$ (increasing by 5) for each topic and take the average over all topics. MAP measures the overall ability of a method to differentiate between known experts and non-experts. The *average precision* (AP) is defined as (Wang et al., 2015):

$$AP = \frac{\sum_{i=1}^n (P@i * \text{rel}(i))}{|R_t|} \quad (14)$$

where i is the rank, $\text{rel}(i)$ is a binary function indicating 1, if the result at i is a known expert, otherwise 0. MAP is the mean value of AP values over all topics, and we use $n = 30$ as used in (Deng et al., 2008).

³ LExR is available to download from http://toinebogers.com/?page_id=240.

⁴ These datasets can be downloaded from <http://www.lbd.dcc.ufmg.br/lbd/collections>.

⁵ These datasets are also no longer publicly available.

⁶ The pre-trained SciBERT model can be downloaded at <https://github.com/allenai/scibert>.

Table 2
Expertise topics and corresponding similar phrases in four datasets.

LExR		IR	
Topic	Phrase	Topic	Phrase
Synthesis	Synthesis	Information retrieval	Information retrieval
Risk factor	Risk factor	Search engine	Search engine
Public health	Public health	Patent search	Patent search
Thin film	Ultrathin film	Data modelling	Information modelling
Development validation	Validation process	Cooperative work	Collaborative working
SW		CL	
Topic	Phrase	Topic	Phrase
Semantic web	Semantic web	Question answering	Question answering
Linked data	Linked data	Knowledge transfer	Knowledge transfer
Text understanding	Text understanding	Summarisation	Summarisation
Auction	Bidding	Machine vision	Computer vision
Invalidity	Inadequacy	Image classification	Image recognition

Table 3
MAP and the improvement ratio of *nVSM*.

	LExR		IR		SW	CL		Avg.		
DLM-0.5	0.200	(233.0%)	0.208	(6.7%)	0.070	(51.4%)	0.063	(68.3%)	0.135	(103.7%)
DLM-0.6	0.159	(318.9%)	0.185	(7.8%)	0.068	(55.9%)	0.058	(82.8%)	0.118	(133.1%)
TFIDF	0.493	(35.1%)	0.206	(7.8%)	0.087	(21.8%)	0.120	(-11.7%)	0.226	(21.7%)
WISER	0.117	(469.2%)	0.150	(48.0%)	0.057	(86.0%)	0.051	(107.8%)	0.094	(192.6%)
<i>nVSM</i>	0.666		0.222		0.106		0.106		0.275	

5.3. Evaluation of *nVSM*

As *nVSM* is one key component in ExpFinder, we first measure its effectiveness. As presented in Section 4.2, the concepts VSM and DLM are closely related. Thus, we compare *nVSM* with TFIDF-based VSM and two particular DLMs: (1) *TFIDF-based VSM* expressed using Eqs. (3) and (7) (denoted as TFIDF); (2) *The DLM model* (Balog et al., 2009; Wang et al., 2015) denoted using Eqs. (3) and (4) in which the probability of individual terms is estimated by Eq. (5), where we use two values for the best λ_θ , 0.5 (*DLM-0.5*) and 0.6 (*DLM-0.6*), as suggested by (Balog et al., 2009) and (Wang et al., 2015), respectively; and (3) A recent probabilistic model *WISER* (Cifariello et al., 2019) that combines the document-centric approach exploiting the occurrence of topics in experts' documents, with the profile-centric approach computing relatedness between experts using an external knowledge source, Wikipedia. Since our work does not consider such an external knowledge source, we only consider *WISER* with the document-centric approach for the fair comparison. In *WISER*, the topic-sensitive weight of an expert x given a topic t is calculated using Reciprocal Rank (Macdonald & Ounis, 2006): $\sum_d \frac{1}{rank(d)}$ that represents the ranks of x 's documents where t appears ($D_{x,t}$). Since t is a phrase, $D_{x,t}$ consists of the subset of D_x that any of t 's constituent terms appears. $rank(d)$ is the ranking position of a document d out of D , where the position is determined by BM25 (Robertson & Zaragoza, 2009). The hyper-parameters $k1$ and b in BM25 are set to be 1.2 and 0.75, respectively, based on the suggestion (Lv & Zhai, 2011).

The evaluation results are presented in Fig. 3 that shows the AP values with n ($n=10, 15, \dots, 30$) of P@ n for all topics. We observe the following: (1) Overall, the VSM approaches (TFIDF and *nVSM*) largely outperform all DLM-0.5, DLM-0.6 and *WISER*. This indicates the VSM approaches can be more effectively used for identifying topic-sensitive experts than the compared DLMs; (2) DLM-0.5 is consistently better than DLM-0.6 but their difference seems minor; and (3) *nVSM* is clearly better than TFIDF from P@10 to P@30 consistently over all the four datasets. Also, *nVSM* substantially outperforms all the compared methods on all the four datasets.

Table 3 shows the results on MAP and the relative improvement ratio of *nVSM* over the other models. The best one in each dataset is denoted in boldface. We see that *nVSM*'s improvements over DLM-0.5 and DLM-0.6 are substantial: up to 318.9% over DLM-0.6 on LE_xR, 7.8% over DLM-0.6 on IR, 55.9% on CL, and 82.8% on CL. Moreover,

nVSM substantially outperforms *WISER* from 48.0% on IR to 469.2% on LE_xR. Also, *nVSM* is highly better than TFIDF except the only one case on CL. On average, we observe that *nVSM* largely outperforms DLM-0.5 in 103.7%; DLM-0.6 in 133.1%; *WISER* in 192.6%; and TFIDF in 21.7% across the four datasets. In summary, the results show an empirical evidence that *nVSM* can be competitive and effectively used for expert finding. Further, these show that ExpFinder is equipped with a powerful component, *nVSM*, for expert finding.

5.4. Finding the best values for personalised parameters in ExpFinder: λ_x and λ_d

We now present how to empirically find the best values for personalised parameters λ_x and λ_d of μ CO-HITS (see Eq. (11)) which is another key component of ExpFinder. Our approach is to make a full use of all the four datasets to determine such values. For this, we measure the *mean* impact of different values of λ_x and λ_d , respectively, on generating the MAP results from the four datasets. Our aim is to provide an empirical guideline for choosing the best values for these parameters. Formally, let Z be the set of candidate values $[0, 0.1, \dots, 1.0]$ for λ_x and λ_d . Then, let us define $MAP(a, b)$ as the MAP value using a pair of $a \in Z$ for λ_x and $b \in Z$ for λ_d . First, we choose the best value for λ_x . To this end, for each value $a \in Z$, we compute the mean of the MAP values with all values in Z in each dataset:

$$Avg(a, \lambda_x) = \frac{1}{|Z|} \sum_{b \in Z} MAP(a, b). \quad (15)$$

Then, we obtain the $|Z|$ -length vector of $Avg(a, \lambda_x)$ for all values in Z . Let us say that this vector is denoted as $Avg(Z, \lambda_x)$. For example, if $Avg(Z, \lambda_x) = [1, 0.9, 0.8, \dots, 0]$, then the corresponding element-wise rank vector is $R(Avg(Z, \lambda_x)) = [11, 10, 9, \dots, 1]$, where the higher rank indicates the higher mean of the MAP values. Similarly, we use $R^i(Avg(Z, \lambda_x))$ to denote the $R(Avg(Z, \lambda_x))$ calculated on the dataset i . Finally, we compute the element-wise mean rank across the four datasets:

$$AvgR(Z, \lambda_x) = \frac{1}{n} \sum_{i=1}^n R^i(Avg(Z, \lambda_x)), \quad (16)$$

where $n = 4$ corresponding to the number of datasets. Using the above equation, we find the best value for λ_x that is the $a \in Z$ generating the highest rank.

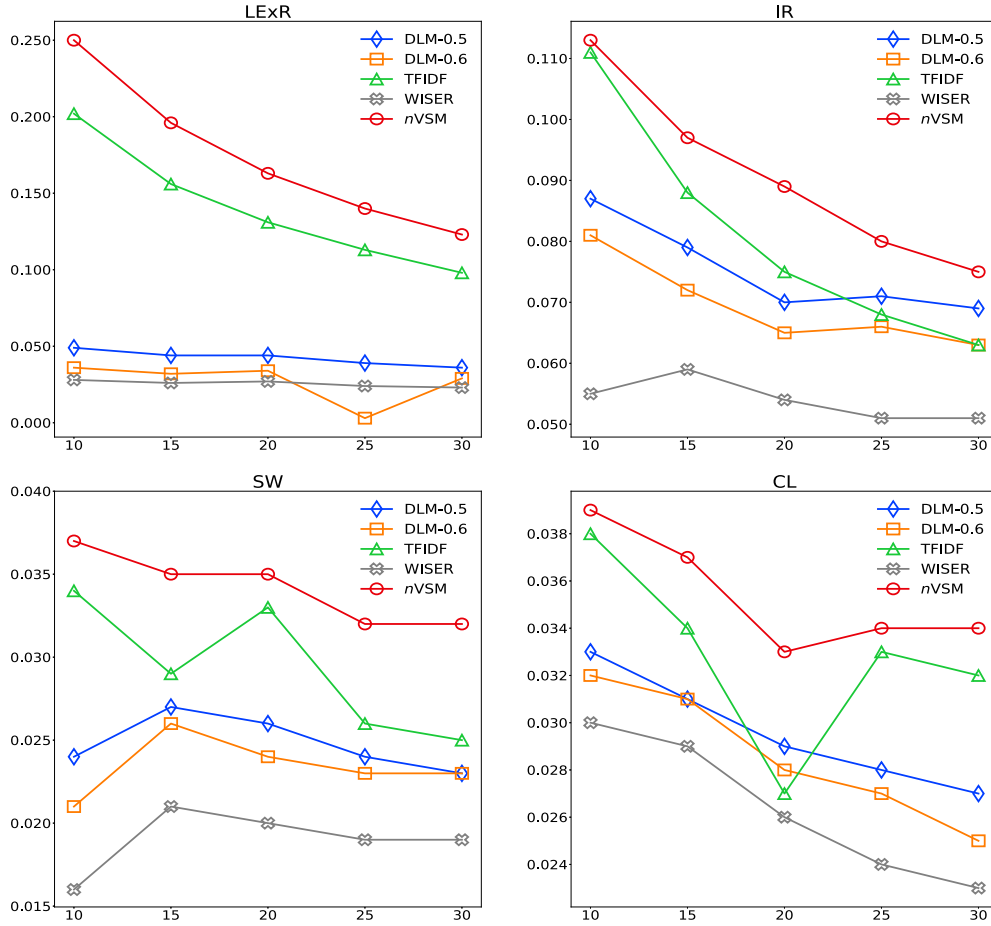


Fig. 3. Comparison on the average precision (AP) values: x-axis shows n of P@n and y-axis shows AP values.

Finding the best value for λ_d is the same as the above procedure, except that we fix a to be the identified best value for λ_x . Thus, Eq. (15) is modified as: $Avg(b, \lambda_d) = MAP(a, b)$. Then, we obtain the $|Z|$ -length vector of $Avg(b, \lambda_d)$ for all possible values for $b \in Z$. This vector is denoted as $Avg(Z, \lambda_d)$. Also, $R^i(Avg(Z, \lambda_d))$ indicates the $R(Avg(Z, \lambda_d))$ calculated on the dataset i . Finally, we compute the element-wise mean rank across the four datasets using the Eq. (16) except that λ_x is replaced with λ_d . By doing so, we find the best value for λ_d by choosing the $b \in Z$ generating the highest rank. Fig. 4(a) - (b) show the average ranks of values in Z for λ_x and λ_d , respectively, across four datasets. As we see, $\lambda_x = 1.0$ produces the highest rank, whereas $\lambda_d = 0.7$ is the highest rank with $\lambda_x = 1.0$. The best ones are denoted in red colour.

5.5. Evaluation of ExpFinder

We now evaluate ExpFinder using the best values for λ_x and λ_d . To measure its relative effectiveness, we also compare it with $nVSM$ as well as three hybrid models: (1) ADT (Gollapalli et al., 2013), (2) Reputation Model (simply RepModel) (Schall, 2015), and (3) CO-HITS (Eq. (10)). Here, for comparison purposes, our focus is to choose three hybrid models that take certain forms of combination of a GM and a VSM. We do not include hybrid models combining a GM and a DLM, as the DLM approaches (e.g., DLM-0.5, DLM-0.6, and WISER) showed lower performance than our VSM approaches (TFIDF (Eq. (7)) and $nVSM$ (Eq. (8))) as presented in Section 5.3. Our objective here is to validate the stronger ability of ExpFinder over those three hybrid models. Finally, we show that ExpFinder works well regardless of topic coverage.

ADT uses an indirect, weighted tripartite (expert–document–topic) graph, where each triplet contains experts, documents and topics.

Experts are connected to their documents, and also documents are connected to the topics based on their occurrences. The weight of an edge between an expert x and a document d (w_{xd}) corresponds to $p(x|d)$ in Eq. (3). The weight of an edge between d and a topic t (w_{dt}) is modelled as $p(t|d)$ in Eq. (3). Recall that we fixed $p(x|d)$ as 1 in Section 5.2. As ExpFinder models $p(t|d)$ as $nTFIDF$, we also model w_{dt} as $nTFIDF$ in ADT for the fair comparison. By combining the tripartite graph and $nTFIDF$, ADT forms a hybrid model (GM + VSM). ADT ranks x given a topic t based on the score function $s(x, t)$ (the higher the more important):

$$s(x, t) = \sum_{d \in D_x} w_{xd} \cdot pweight(d, t), \quad (17)$$

where $pweight(d, t) = \sum_{p \in P(d, t)} \prod_i w(e_i)$ where p is a path between d and t comprising of edges such that $p = e_1 e_2 \dots e_n$; $P(d, t)$ is the set of all possible paths between d and t ; and $w(e_i)$ is the weight of the i th edge in p .

RepModel (Schall, 2015) was originally designed to estimate the topic-sensitive reputation of an organisation in the context of scientific research projects. This model uses topic-sensitive CO-HITS given a topic, where an organisation is seen as an expert and a project is seen as a document in our work. Thus, using the CO-HITS notations in Eq. (10), RepModel models $A(x; t)^k$ as $\sum_{w \in t} w_w A(x; w)^k$ and $H(d; t)^k$ as $\sum_{w \in t} w_w H(d; w)^k$, where $w_w = \frac{1}{|t|}$. As λ_x and λ_d , we use 0.85 as used in Schall (2015). In RepModel, the personalised weights $\alpha_{x;w}$ and $\alpha_{d;w}$ are defined as: $\alpha_{d;w} = tf(d, w)$ denoting the term frequency of w divided by $|d|$; and $\alpha_{x;w} = s(x, w)$ if w appears in D_x , and 0, otherwise. $s(x, w)$ is defined as $1 - \frac{\max_s - weight(x, w)}{\max_s - \min_s}$, where \max_s and \min_s are the max and min values of $weight_i(x, w)$ for all experts, and $weight(x, w)$ is the weight of x on w calculated by the number of documents in D_x .

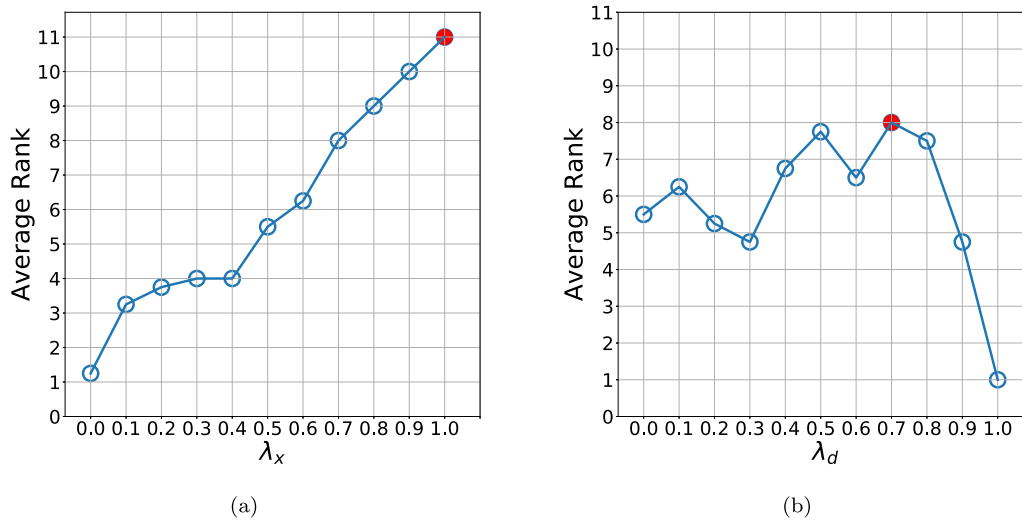


Fig. 4. Finding the best values for λ_x and λ_d .

Table 4

MAP and the improvement ratio of ExpFinder.

	LExR	IR	SW	CL	Avg.
<i>n</i> VSM	0.666 (12.1%)	0.222 (15.3%)	0.106 (7.6%)	0.106 (4.7%)	0.275 (11.6%)
ADT	0.574 (30.1%)	0.186 (37.6%)	0.077 (48.1%)	0.106 (4.7%)	0.236 (30.1%)
RepModel	0.260 (187.3%)	0.144 (77.8%)	0.061 (86.9%)	0.070 (58.6%)	0.134 (129.1%)
CO-HITS	0.611 (22.3%)	0.228 (12.3%)	0.104 (9.6%)	0.088 (26.1%)	0.258 (19.0%)
ExpFinder	0.747	0.256	0.114	0.111	0.307

where w appears. Thus, by combining CO-HITS and the above tf -based weighting schemes, RepModel forms a hybrid model (GM + VSM). We also set $k=5$ as done for μ CO-HITS.

For the fair comparison between μ CO-HITS and CO-HITS, personalised weights $\alpha_{x;t}$ and $\alpha_{d;t}$ in CO-HITS in Eq. (10) are set as $\mathbf{TX}_{i(t),i(x)}$ and $\mathbf{TD}_{i(t),i(d)}$, respectively, as μ CO-HITS. Following the same experiment in Section 5.4, we found that the best values for λ_x and λ_d are chosen as 1.0 and 1.0, respectively, for CO-HITS. We also fix k as 5 in Eq. (10) as ExpFinder. In our comparison below, CO-HITS indicates an alternative ExpFinder form incorporating *n*VSM into CO-HITS. Thus, CO-HITS forms a hybrid model (GM + VSM).

Fig. 5 shows the evaluation results based on the AP values with n ($n=10, 15, \dots, 30$) of P@ n for all topics. First, when comparing ExpFinder with ADT, although ADT is slightly better than ExpFinder over two datasets IR and CL at $n=10$, ExpFinder largely outperforms at all values for n of P@ n , i.e. $n=15, \dots, 30$, on all the datasets. It is also clear that ExpFinder is substantially better than RepModel on all x -axis values. Second, ExpFinder is consistently better than CO-HITS on LExR and SW, and very similar to each other on IR. On CL, CO-HITS is better than at $n=10$ and $n=15$, but similar at $n=20$, and worse than ExpFinder at $n=25$ and $n=30$. Overall, these results also show that μ CO-HITS can have a competitive potential for improving the performance over CO-HITS. Third, to determine the impact of incorporating *n*VSM into μ CO-HITS in ExpFinder, let us compare ExpFinder with *n*VSM. As observed, ExpFinder is clearly and consistently better than *n*VSM over LExR and SW, although they are similar on IR and *n*VSM looks better than ExpFinder on CL. On average, we observe that our hybrid model ExpFinder combining *n*VSM with μ CO-HITS is observed to be more powerful than only *n*VSM.

Table 4 shows the evaluation results in MAP. The best one is denoted in boldface. As observed, ExpFinder outperforms all the methods in 11.6%, 30.1%, 129.1% and 19.0% over *n*VSM, ADT, RepModel and CO-HITS, respectively, on average. Interestingly, *n*VSM is observed as the second better one. This also shows that our *n*VSM for expert finding is more competitive than the compared GMs.

Finally, it may be also worth analysing the distribution of the MAP values of a model across topics based on *topic coverage* in each dataset. In our context, the topic coverage of a topic t means the number of known experts having expertise t . This enlightens how the model particularly works better or worse at which topic coverage values. Intuitively, it may be harder to find experts for topics whose topic coverage is lower. For this analysis, we pay only attention to our two models *n*VSM and ExpFinder. By comparing their distributions, we can identify which model is better than the other on what topic coverage values.

The analysis results are seen in Fig. 6. In each plot, each value on the x -axis shows a topic coverage value. Each value on the y -axis shows the MAP value of the set of topics with the same topic coverage. For example, on LExR, the value 5 on the x -axis means that the topic coverage is 5. The corresponding y -axis value 0.85 indicates that the MAP value of the set of topics with the topic coverage 5 is 0.85. Each MAP value is also calculated using $n=30$ of P@ n . Each black circle represents that both of *n*VSM and ExpFinder have the same MAP value. We can observe the following: (1) On LExR, ExpFinder dominantly outperforms or is equal to *n*VSM over all the topic coverage values except the 4 cases (i.e. 13, 15, 18, and 19); (2) On IR, ExpFinder also shows its improvement over *n*VSM over most of the topic coverage values, while one MAP value is the same on the topic coverage 26; (3) On SW, we observe that ExpFinder prevalingly outperforms *n*VSM across most of the topic values as ExpFinder's distribution is predominantly higher than *n*VSM's distribution; (4) On CL, although *n*VSM is better than ExpFinder on the topic coverage values 10 and 30, ExpFinder is observed notably better than *n*VSM over the other topic coverage values; and (5) there seems no clear clue that ExpFinder performs particularly better on which range of topic coverage. For example, on LExR, ExpFinder seems to perform better in smaller topic coverage values, as the MAP values from 0 to 5 are clearly higher than those from 6 to 15 on the axis. But this pattern is not consistent with the datasets on SW and CL, where ExpFinder performs better on larger topic coverage values. In conclusion, the above observations may indicate that overall ExpFinder outperforms *n*VSM regardless of topic coverage, showing the validity of the design

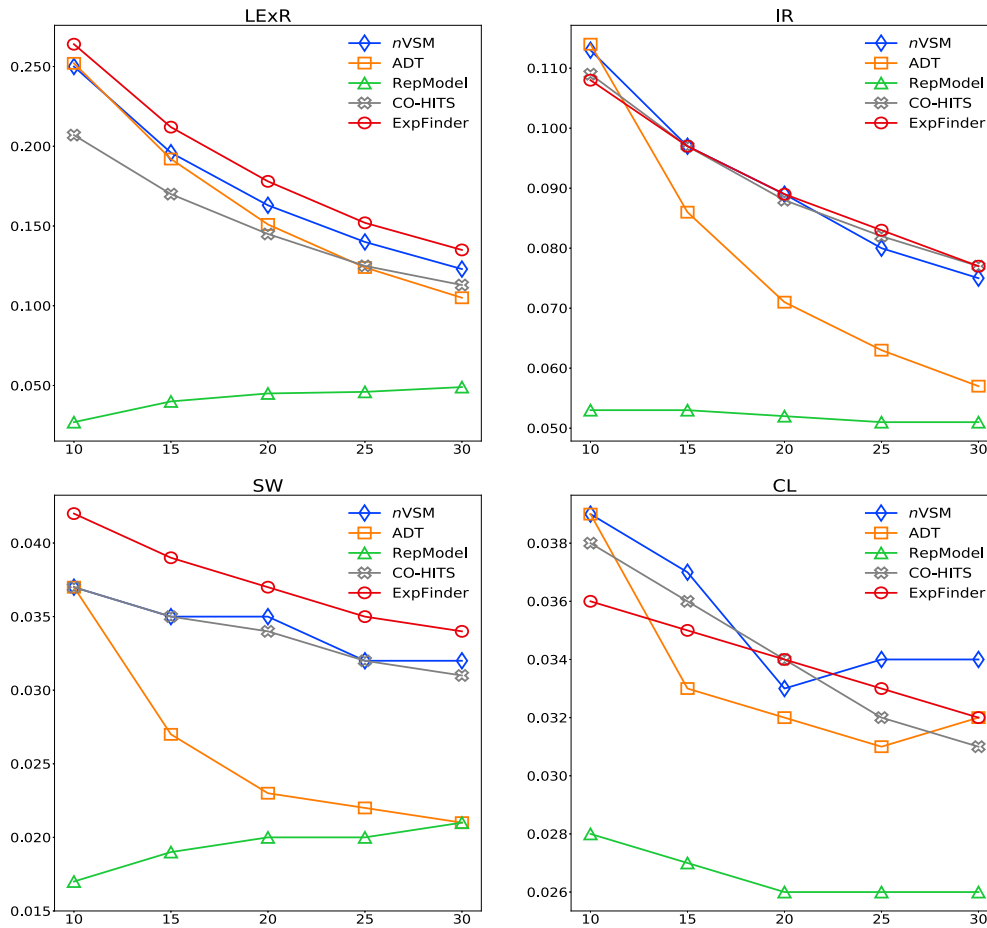


Fig. 5. Comparison on the average precision (AP) values: x-axis shows n of $P@n$ and y-axis shows AP values.

paradigm of ExpFinder, that is, incorporating n VSM into μ CO-HITS can have powerful capability for expert finding.

5.6. Discussions and future work

Using the four datasets from academic domains, we evaluated ExpFinder and its two key components n VSM and μ CO-HITS, and compared the results with other expert finding models: TFIDF based VSM (denoted as TFIDF), DLM-0.5 (Balog et al., 2009) and DLM-0.6 (Wang et al., 2015), WISER (Cifariello et al., 2019) as well three hybrid expert finding models that form the combination of a GM and a VSM: ADT (GM + n TFIDF), RepModel (GM + the tf -based weighting schemes) and CO-HITS (GM + n VSM).

We showed the capability of n VSM using different AP values ($n=10, 15, \dots, 30$) and MAP, in comparison with TFIDF, DLM-0.5 and DLM-0.6. On average, the improvement ratio of n VSM over them was turned out as from 21.7% to 133.1% in MAP. We also presented the empirical method for finding the best values of the two parameters used in ExpFinder, λ_x and λ_d , based on the ranking of the MAP values. Moreover, we showed how much ExpFinder performs better than all the compared methods in Tables 3 and 4 in MAP. It was demonstrated that ExpFinder improves DLM-0.5 and DLM-0.6 in 127.5% and 160.2%, respectively; TFIDF in 35.9%; WISER in 71.5%; ADT in 31%; n VSM in 11.6%; RepModel in 129.1%; and CO-HITS in 19.0%. Further, we showed that ExpFinder incorporating n VSM into μ CO-HITS indeed improves a pure VSM approach n VSM. It means that exploiting network propagation effects on ECG using μ CO-HITS with the outcome of n VSM can contribute to better estimating topic-sensitive weights of experts. Also, by comparing ExpFinder with CO-HITS, we proved that the

proposed variation schemes (i.e., dynamic personalised weighting and the average-based aggregation of authority and hub scores) embedded in μ CO-HITS contribute to improving the accuracy of the pure CO-HITS algorithm (i.e. fixed personalised weighting and sum-based aggregation of authority and hub scores) (Deng et al., 2009). Finally, we analysed that ExpFinder works well regardless of topic coverage values. Our all evaluation results reinforce our motivation of designing ExpFinder that the proposed hybrid model ExpFinder for expert finding is effective and competitive for expert finding.

As future work, it could be worth to investigate ways for improving precision for expert finding. As we have observed in Table 4, the average MAP value of ExpFinder is 0.307 across the four datasets. In the literature, we can also observe the similar MAP results. For example, WISER (Cifariello et al., 2019) reported that its best MAP values are 0.214 and 0.363 on the two datasets, BMExpert (Wang et al., 2015) also showed 0.06 as the best MAP value of the DLM (Balog et al., 2009) on the single dataset, and ADT (Gollapalli et al., 2013) also showed its best MAP values are 0.0943 and 0.1986 on the two datasets. We also plan to accommodate a general expertise knowledge source as (Cifariello et al., 2019), e.g. Wikipedia, into ExpFinder to see its potential for enhancing ExpFinder's capability. Another interesting future work would be to examine graph embedding techniques for expert finding. One idea would be that we extend ECG by constructing an expert-document-topic graph based on their semantic relationships. Then we can train a machine to transform nodes, edges and their features into a vector space while maximally preserving their relationship information. Once we would be successfully able to map such a graph to a vector space, we could estimate the importance of experts given documents or topics by measuring their similarity (or relevance) between experts and documents or between experts and topics in terms of their corresponding

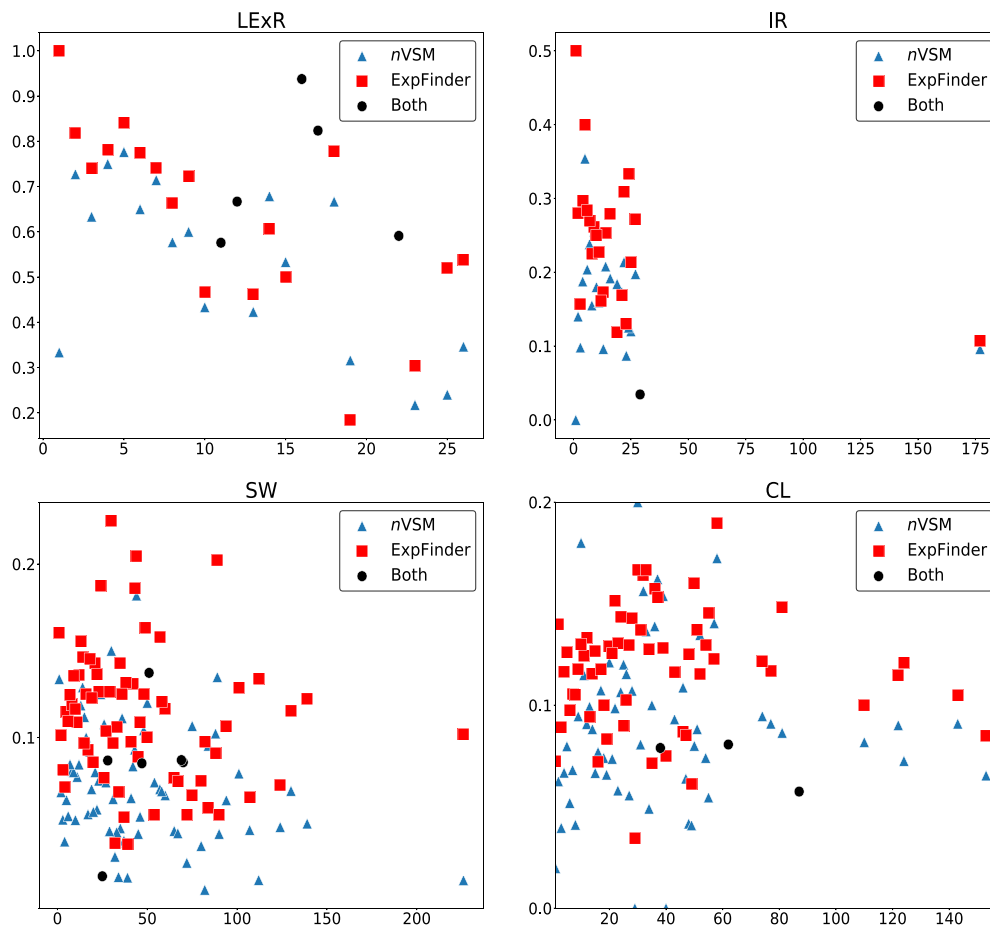


Fig. 6. MAP values of *nVSM* and ExpFinder: the *x*-axis shows topic coverage values, and the *y*-axis shows the MAP values (topic coverage of a topic *t* means the number of known experts associated with *t*).

vector values. Finally, we could enhance evaluations of expert finding methods by measuring their run-time efficiency to estimate the temporal and spatial complexity of the methods. This could bring insight into accuracy of expert finding methods as well as their computational costs.

6. Conclusion

In this paper, we proposed ExpFinder a novel hybrid model for expert finding. We presented the design of ExpFinder and conducted comprehensive empirical experiments to evaluate and validate its effectiveness using four publicly accessible datasets (LExR (Mangaravite et al., 2016), Information Retrieval, Semantic Web and Computational Linguistics in DBLP dataset (Bordea et al., 2013)) from the academic domains. The novelty of ExpFinder is in its incorporation of a novel *N*-gram vector space model (*nVSM*) into μ CO-HITS. The key to designing *nVSM* is to utilise the state-of-the-art IDF method (Shirakawa et al., 2017) for estimating the topic-sensitive weights of experts given a topic. Such estimated weights are further improved by incorporating them into ECG using μ CO-HITS. Our novelty of μ CO-HITS is to design two variation schemes of CO-HITS (Deng et al., 2009), thus proposing a unified formula for successfully integrating *nVSM* with μ CO-HITS. We conduct a comprehensive evaluation study to demonstrate that ExpFinder is highly competitive expert finding model, in comparison with six different state-of-the art expert finding models.

CRedit authorship contribution statement

Yong-Bin Kang: Investigation, Conceptualization, Methodology, Software, Formal analysis, Writing – review & editing. **Hung Du:**

Methodology, Software, Formal analysis, Writing – review & editing. **Abdur Rahim Mohammad Forkan:** Methodology, Writing – review & editing. **Prem Prakash Jayaraman:** Methodology, Writing – review & editing. **Amir Aryani:** Methodology, Writing – review & editing. **Timos Sellis:** Project administration, Funding acquisition, Supervision, Investigation, Conceptualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared my code and data in the manuscript.

References

- Alhabashneh, O., Iqbal, R., Doctor, F., & James, A. (2017). Fuzzy rule based profiling approach for enterprise information seeking and retrieval. *Information Sciences*, 394, 18–37.
- Aslay, C., O'Hare, N., Aiello, L. M., & Jaimes, A. (2013). Competition-based networks for expert finding. In *SIGIR '13, Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval* (pp. 1033–1036).
- Balog, K., Azzopardi, L., & de Rijcke, M. (2009). A language modeling framework for expert finding. *Information Processing & Management*, 45, 1–19.
- Beltagy, I., Lo, K., & Cohan, A. (2019). Scibert: Pretrained language model for scientific text. In *EMNLP*.
- Bok, K., Jeon, I., Lim, J., & Yoo, J. (2019). Expert finding considering dynamic profiles and trust in social networks. *Electronics*, 8(1165).

- Bordea, G., Bogers, T., & Buitelaar, P. (2013). Benchmarking domain-specific expert search using workshop program committees. In *Proceedings of the 2013 workshop on computational scientometrics* (pp. 19–24).
- Campbell, C. S., Maglio, P. P., Cozzi, A., & Dom, B. (2003). Expertise identification using email communications. In *Proceedings of the twelfth international conference on information and knowledge management* (pp. 528–531).
- Chuang, C. T., Yang, K. H., Lin, Y. L., & Wang, J. H. (2014). Combining query terms extension and weight correlative for expert finding. In *2014 IEEE/WIC/ACM international joint conferences on web intelligence and intelligent agent technologies. Vol. 1* (pp. 323–326). IEEE.
- Cifariello, P., Ferragina, P., & Ponza, M. (2019). WISER: A semantic approach for expert finding in academia based on entity linking. *Information Systems*, 82, 1–16.
- Deng, H., King, I., & Lyu, M. R. (2008). Formal models for expert finding on DBLP bibliography data. In *ICDM '08, Proceedings of the 2008 eighth IEEE international conference on data mining* (pp. 163–172).
- Deng, H., King, I., & Lyu, M. R. (2011). Enhanced models for expertise retrieval using community-aware strategies. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 42, 93–106.
- Deng, H., Lyu, M. R., & King, I. (2009). A generalized CO-HITS algorithm and its application to bipartite graphs. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 239–248).
- Faisal, M. S., Daud, A., Akram, A. U., Abbasi, R. A., Aljohani, N. R., & Mehmood, I. (2019). Expert ranking techniques for online rated forums. *Computers in Human Behavior*, 100, 168–176.
- Gollapalli, S. D., Mitra, P., & Giles, C. L. (2013). Ranking experts using author-document-topic graphs. In *JCDL '13, Proceedings of the 13th ACM/IEEE-CS joint conference on digital libraries* (pp. 87–96).
- Gonçalves, R., & Dorneles, C. F. (2019). Automated expertise retrieval: A taxonomy-based survey and open issues. *ACM Computing Surveys*, 52, 1–30.
- Han, J., Teng, X., & Cai, X. (2019). A novel network optimization partner selection method based on collaborative and knowledge networks. *Information Sciences*, 484, 269–285. <http://dx.doi.org/10.1016/j.ins.2019.01.072>.
- Husain, O., Salim, N., Alias, R. A., Abdelsalam, S., & Hassan, A. (2019). Expert finding systems: A systematic review. *Applied Sciences*, 9(4250).
- Jiang, X., Sun, X., Yang, Z., Zhuge, H., & Yao, J. (2016). Exploiting heterogeneous scientific literature networks to combat ranking bias: Evidence from the computational linguistics area. *Journal of the Association for Information Science and Technology*, 67, 1679–1702.
- Kang, Y.-B., Delir Haghghi, P., & Burstein, F. (2014). Cfinder: An intelligent key concept finder from text for ontology development. *Expert Systems with Applications*, 41, 4494–4504.
- Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46, 604–632.
- Koumenides, C. L., & Shadbolt, N. R. (2014). Ranking methods for entity-oriented semantic web search. *Journal of the Association for Information Science and Technology*, 65, 1091–1106.
- Kundu, D., & Mandal, D. P. (2019). Formulation of a hybrid expertise retrieval system in community question answering services. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, 49, 463–477.
- Liu, D.-R., Chen, Y.-H., Kao, W.-C., & Wang, H.-W. (2013). Integrating expert profile, reputation and link analysis for expert finding in question-answering websites. *Information Processing & Management*, 49, 312–329.
- London, A., & Csendes, T. (2013). Hits based network algorithm for evaluating the professional skills of wine tasters. In *2013 IEEE 8th international symposium on applied computational intelligence and informatics* (pp. 197–200). <http://dx.doi.org/10.1109/SACI.2013.6608966>.
- Lu, K. (2013). An insight into vector space modeling and language modeling. In *ICConference*. <http://dx.doi.org/10.9776/13332>.
- Lv, Y., & Zhai, C. (2011). Adaptive term frequency normalization for BM25. In *Proceedings of the 20th ACM international conference on information and knowledge management* (pp. 1985–1988).
- Ma, Z., Liu, Y., Hu, Y., Yang, J., Liu, C., & Dai, H. (2022). Efficient maintenance for maximal bicliques in bipartite graph streams. *World Wide Web*, 25, 857–877. <http://dx.doi.org/10.1007/s11280-021-00927-x>.
- Macdonald, C., & Ounis, I. (2006). Voting for candidates: Adapting data fusion techniques for an expert search task. In *Proceedings of the 15th ACM international conference on information and knowledge management* (pp. 387–396).
- Mangaravite, V., & Santos, R. L. (2016). On information-theoretic document-person associations for expert search in academia. In *SIGIR '16, Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval* (pp. 925–928).
- Mangaravite, V., Santos, R. L., Ribeiro, I. S., Gonçalves, M. A., & Laender, A. H. (2016). The LEXR collection for expertise retrieval in academia. In *Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval* (pp. 721–724).
- Riahi, F., Zolaktaf, Z., Shafiei, M., & Milios, E. (2012). Finding expert users in community question answering. In *Proceedings of the 21st international conference on world wide web* (pp. 791–798).
- Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3, 333–389. <http://dx.doi.org/10.1561/1500000019>.
- Roelleke, T., & Wang, J. (2008). Tf-idf uncovered: A study of theories and probabilities. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 435–442).
- Schall, D. (2015). *A social network-based recommender systems*. Springer.
- Schunk, L. K., & Cong, G. (2010). Using transactional data from erp systems for expert finding. In *Database and expert systems applications* (pp. 267–276).
- Shirakawa, M., Hara, T., & Nishio, S. (2017). IDF for word N-grams ACM trans. *Information Systems*, 36.
- Stankovic, M., Wagner, C., Jovanovic, J., & Laublet, P. (2010). Looking for experts? What can linked data do for you? In *Proceedings of the WWW workshop on linked data on the web*.
- Sziklai, B. (2018). How to identify experts in a community? *International Journal of Game Theory*, 47, 155–173.
- Torkzadeh Mahani, N., Dehghani, M., Mirian, M. S., Shakery, A., & Taheri, K. (2018). Expert finding by the dempster-shafer theory for evidence combination. *Expert Systems*, 35.
- Truong, B. T., Allen, O. M., & Menczer, F. (2022). News sharing networks expose information polluters on social media. <http://dx.doi.org/10.48550/ARXIV.2202.00094>.
- Van Gysel, C., de Rijke, M., & Worring, M. (2016). Unsupervised, efficient and semantic expertise retrieval. In *Proceedings of the 25th international conference on world wide web* (pp. 1069–1079).
- Wang, B., Chen, X., Mamitsuka, H., & Zhu, S. (2015). Bmexpert Mining medline for finding experts in biomedical domains based on language model. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 12, 1286–1294.
- Wang, G. A., Jiao, J., Abrahams, A. S., Fan, W., & Zhang, Z. (2013). Expertrank: A topic-aware expert finding algorithm for online knowledge communities. *Decision Support Systems*, 54, 1442–1451.
- Yang, K.-C., Aronson, B., & Ahn, Y.-Y. (2020). BiRank: Fast and flexible ranking on bipartite networks with R and python. <http://dx.doi.org/10.5281/zenodo.3939045>.
- Yeniterzi, R., & Callan, J. (2014). Constructing effective and efficient topic-specific authority networks for expert finding in social media. In *Proceedings of the first international workshop on social media retrieval and analysis* (pp. 45–50).
- Zhang, J., Ackerman, M. S., & Adamic, L. (2007). Expertise networks in online communities: structure and algorithms. In *Proceedings of the 16th international conference on world wide web* (pp. 221–230).