ExpFinder: An Ensemble Expert Finding Model Integrating N-gram Vector Space Model and μCO-HITS

Yong-Bin Kang, Hung Du, Abdur Rahim Mohammad Forkan, Prem Prakash Jayaraman, Amir Aryani, Timos Sellis, *Fellow, IEEE*

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 expertise data makes identifying the right experts a challenging problem. Existing approaches for finding experts given a topic can be categorised into information retrieval techniques based on vector space models, document language models, and graph-based models. In this paper, we propose *ExpFinder*, a new ensemble model for expert finding, that integrates a novel N-gram vector space model, denoted as nVSM, and a graph-based model, denoted as *μCO-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 μCO-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%.

Index Terms—ExpFinder, Expert finding, N-gram Vector Space Model, μCO-HITS, Expert collaboration graph

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. In general, an *expert* is defined as a person who has sufficient knowledge and skills in a given field [1]. Such knowledge and skills are called *expertise*. While digitally available data (e.g. scientific publications) describing expertise of experts 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 term these as *documents* in this paper) such as scientific publications [2]. Based on the literature [3], 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 first task (i.e. expert finding) and propose an ensemble model for it from unstructured documents. We use the term *topic* to represent a field of expertise.

Most existing approaches for expert finding are based on *vector space models* (VSM), *document language models* (DLM), or *graph-based models* (GM). In VSM, expert finding is often solved by modeling the weights of topics, associated with the documents produced by experts, using Term Frequency-Inverse Document Frequency (TFIDF) or its variation [4, 5]. In DLM, expert finding is achieved by estimating the probability that a topic would be observed in the documents of an expert [6, 7, 8]. In GM, a graph is 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 analytics such as expert-document-term association paths [9], Hyper-Induced Topic Search (HITS) [10, 11], or social network based link analysis methods [12, 13, 14].

Although various models in these approaches aforementioned have been proposed, integrating VSM and GM for expert finding has been little studied in the

- *Yong-Bin Kang is with Department of Media and Communication, Swinburne University of Technology, Australia* ***E-mail:*** *ykang@swin.edu.au*
- *Hung Du, Abdur Rahim Mohammad Forkan, Prem Prakash Jayaraman, and Timos Sellis are with Department of Computer Science and Software Engineering, Swinburne University of Technology, Australia* ***E-mail:*** *hungdu@swin.edu.au, fforkan@swin.edu.au, pjayaraman@swin.edu.au, and tsellis@swin.edu.au*
- *Amir Aryani is with Social Innovation Research Institute, Faculty of Business and Law, Swinburne University of Technology, Australia.* ***E-mail:*** *aaryani@swin.edu.au*

literature. In this work, we propose an ensemble model for expert finding, *ExpFinder*[1], that integrates a novel N-gram VSM, denoted as nVSM, with a GM using an *expert collaboration graph (ECG)*. We develop nVSM for estimating the expertise degree (or *weight*) of an expert given a topic by leveraging the recent Inverse Document Frequency (IDF) weighting [15] 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 in the form of an expert-document bipartite graph to represent the associations between experts and documents based on the co-authorship information. To estimate the weight of an expert given a topic on the ECG, we propose the GM, μCO-HITS, that is formed by applying two variation schemes to the generalised CO-HITS [16] algorithm.

This paper makes three main contributions. First, to our best knowledge, ExpFinder is the first attempt to introduce nVSM for expert finding. Second, we propose ExpFinder an ensemble model that combines nVSM and the GM using μCO-HITS to create a stronger model for expert finding that achieves better performance than a single one. ExpFinder incorporates the weights of experts estimated by nVSM into an ECG and uses μCO-HITS on the ECG to better estimate the weights of experts for a given topic. Third, we conduct comprehensive empirical evaluations to measure the effectiveness of ExpFinder using four different datasets (LExR [17] and three DBLP datasets [18]) in academic domains and compare the results with six different expert finding models: the TFIDF-based VSM, two DLMs [6, 8] and three GMs [9, 19, 16].

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 [3]. 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 $\mathcal{D}_x$. That is, the weight of an expert x given a topic t is estimated by aggregating relevance scores between each document in $\mathcal{D}_x$ 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 [20]. A variation of TFIDF was also applied for expert finding in an organization's ERP system [21]. The work [4] 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 [5].

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 [6], or between an expert and its publications [22]. BMExpert [7] used the DLM [6] for expert finding using three factors: relevance of documents to the topic, importance of documents, and associations between documents and experts. Similarly, the work [23] 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* [8], estimated the importance of experts' documents given a topic using BM25 [24]. 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) [10, 11, 25] and PageRank [26]. 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 [27]. Also, a modified PageRank algorithm was developed and applied for finding experts in online knowledge communities [28]. 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 [10] 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 [16] 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 [16]. Using an author-document-topic (ADT) graph, the expert finding GM model [9] leveraged possible paths

1. ExpFinder's source code is publicly available on https://github.com/Yongbinkang/ExpFinder

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 [12] 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 [13] 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 among DLM, VSM, and/or GM [29]. For example, AuthorRank [30] combined a generative probabilistic DLM and a PageRank-like GM based on community engagement of expert candidates. The DLM was used to identify the most relevant documents, while the GM was used to model the authors’ authorities based on the community co-authorship. The work [31] combined a cluster-based language model and a VSM for finding experts in question and answer communities. The authors [29] proposed a complex model for community question answering using a variation of the DLM [6] and a HITS-based GM (the HITS algorithm on a competition based expertise network [32]), where the scores from these models were linearly combined to rank experts given a question. The work [33] used the Dempster-Shafer combination theory to combine the DLM [6] and a graph algorithm that analyses a social interaction of experts. However, this work did not provide technical details on how such combination is done.

Differing from the above approaches, ExpFinder is a first attempt in devising an ensemble model that incorporates nVSM into μCO-HITS on an ECG. The proposed nVSM takes advantage of the IDF weighting for N-grams [15]. μCO-HITS is a novel variation of CO-HITS and runs on an ECG (i.e. expert-document bipartite graph), compared to previous works [10, 11] that applied HITS on an expert-expert graph.

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 [34]. 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

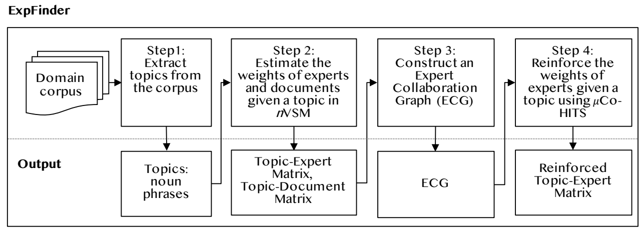


Fig. 1: The overview of ExpFinder.

information in the proposed nVSM. The second facet is μCO-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 nVSM into the ECG, and reinforce the weights of experts given a topic using μCO-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 nVSM. 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 nTFIDF lies in a combination of the frequency of the topic with the IDF method of N-grams over the corpus [15]. 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 μCO-HITS: As presented above, to rank experts, ExpFinder integrates the two knowledge facets: (1) nVSM to estimate the weights of the experts and documents given a topic (**Step 2**); and (2) μCO-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 labeled 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 labeled 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 labeled 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.

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 $\mathcal{D}$. First, for each document $d \in \mathcal{D}$, 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:

$$(\mathrm{JJ})^*|(\mathrm{VBN})^*|(\mathrm{VBG})^*(\mathrm{N})^+, \tag{1}$$

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 nTFIDF in nVSM. These matrices will be used as the input to μCO-HITS.

4.2.1 *Topic-Expert Matrix Creation*

To create a $\mathbf{TX}$, our fundamental is to utilise the definition of the DLM [6, 7] 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 [6]:

$$p(x|t) = p(x,t)/p(t), \tag{2}$$

where $p(x,t)$ is the joint probability of x and t, and $p(t)$ is the probability of t. We can 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 $\mathcal{D}_x$ [6]:

$$\begin{aligned} p(x,t) &= \sum_{d \in \mathcal{D}_x} p(x,d,t) = \sum_{d \in \mathcal{D}_x} p(d)p(x,t|d) \\ &= \sum_{d \in \mathcal{D}_x} p(d)p(t|d)p(x|d). \end{aligned} \tag{3}$$

In Eq. 3, we observe the following notations [7]:
- $p(d)$ 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 [7]).
- $p(t|d)$ is the conditional probability of t given d.

In the DLM [6, 7], 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 [6, 7]:

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

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:

$$p(w|\theta_d) = (1-\lambda_\theta)p(w|d) + \lambda_\theta p(w|\mathcal{D}) \tag{5}$$

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 $p(w|\mathcal{D})$ is the term frequency of w in $\mathcal{D}$ normalised by $|\mathcal{D}|$, i.e., $tf(w,|\mathcal{D}|)$. The parameter λ_θ controls the influence of the two probabilities.

We now present our novelty for estimating $p(t|d)$ using nTFIDF. Since nTFIDF 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 [35]. 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 $\mathcal{D}$. 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:

$$-\sum_w p_w \log q_w = \sum_w p_w \log \frac{1}{q_w}, \tag{6}$$

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:

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

Thus, as highlighted in [36], 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 ($tf(w,|\mathcal{D}|)$: term frequency of w in $\mathcal{D}$ normalised by $|\mathcal{D}|$.

Based on the above observation, we now present our approach for estimating $p(t|d)$ using nTFIDF in nVSM. Although some variant forms of TFIDF methods have been proposed, the majority of TFIDF methods use the same IDF function [15]. 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 [15]. 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 nVSM is the proposed formula nTFIDF 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 nTFIDF, $p(t|d)$ is defined as:

$$\begin{aligned} p(t|d) &\approx n\mathrm{TFIDF}(t,d) = ntf(t,d) \cdot nidf(t), \text{ where} \\ ntf(t,d) &= \frac{\sum_{i=1}^{n} tf(w_i,d)}{|t|}, \\ nidf(t) &= \log \frac{|\mathcal{D}| \cdot df(t)}{df(w_1 \wedge w_2 \wedge \ldots \wedge w_n)^2} \end{aligned} \tag{8}$$

where $w_1,\ldots,w_n$ 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 [15]. Eq. 8 applies for all $|t| \geq 1$, where $nidf(t)$ is equal to the log-IDF, $\log \frac{|\mathcal{D}|}{df(t)}$, in Eq. 7, when $|t|$=1.

Finally, in nVSM, $p(x|t)$ in Eq. 3 is calculated using $p(t|d)$ in Eq. 8 and is stored into the entry $\mathbf{TX}_{i(t),i(x)}$, 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 [6] again, we can 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:

$$p(d|t) = p(t|d)p(d)/p(t), \tag{9}$$

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 nVSM, one potential limit may be that $p(x|t)$ in Eq. 3 mainly relies on the documents $\mathcal{D}_x$ (i.e. $\sum_{d \in \mathcal{D}_x}$), 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 μCO-HITS over the ECG. More specifically, ExpFinder incorporates the topic-sensitive weights of experts given topic, estimated by

nVSM, into an ECG and reinforces such weights using μCO-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 $\mathcal{D}$ (also called *hubs*). Thus, the node set $V = \mathcal{X} \cup \mathcal{D}$. 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 $d \in \mathcal{D}$ 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 $\mathcal{D}$ to $\mathcal{X}$. Given e_{dx}, its weight, denoted as w_{dx}, comes from $\mathbf{DX}_{i(d),i(x)}$ (see Section 3.2).

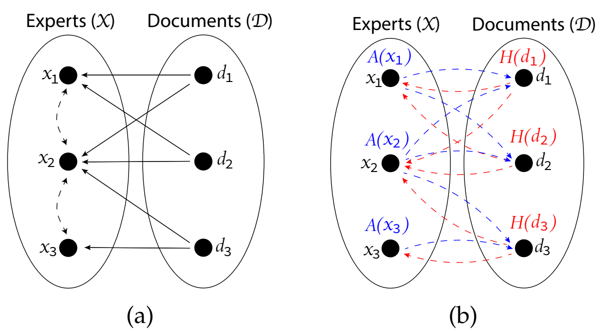


Fig. 2: An example ECG (a) and the score propagation of authority and hub nodes (b).

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 μCO-HITS

Note that μCO-HITS is a variation of CO-HITS [16]. 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 nVSM, using μCO-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 $\mathcal{D}$ 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 μCO-HITS, we take two steps. First, we extend the CO-HITS equation [16] 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 nVSM, 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:

$$
\begin{aligned}
A(x;t)^k &= (1-\lambda_x)\alpha_{x;t} + \lambda_x \sum_{e_{dx}\in E} w_{dx} H(d;t)^{k-1} \\
H(d;t)^k &= (1-\lambda_d)\alpha_{d;t} + \lambda_d \sum_{e_{du}\in E} w_{du} A(u;t)^k
\end{aligned}
\tag{10}
$$

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.
- 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 staring 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 [37], that is, $\left(\sum_{x_i\in\mathcal{X}} \alpha_{x_i;t}\right)^{1/2} = 1$ and $\left(\sum_{d_i\in\mathcal{D}} \alpha_{d_i;t}\right)^{1/2} = 1$. 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 [16]. 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 person-

alised 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 μCO-HITS equation and apply it on the underlying ECG using two variation schemes of topic-sensitive CO-HITS:

$$A(x;t)^k = (1-\lambda_x)A(x;t)^{k-1} + \lambda_x \left(\frac{\sum\limits_{e_{dx}\in E} w_{dx} H(d;t)^{k-1}}{\sum\limits_{e_{dx}\in E} w_{dx}} \right)$$
$$H(d;t)^k = (1-\lambda_d)H(d;t)^{k-1} + \lambda_d \left(\frac{\sum\limits_{e_{du}\in E} w_{du} A(u;t)^{k}}{\sum\limits_{e_{du}\in E} w_{du}} \right) \quad (11)$$

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 $\alpha_{d;t}$, μCO-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 μCO-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 nTFIDF in nVSM. 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 nVSM with μCO-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 nVSM approach. Note that nVSM 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 μCO-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 nVSM. Note that in nVSM, 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 μCO-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 μCO-HITS, 'μ' indicates the 'average' so that μCO-HITS means a particular topic-specific CO-HITS using the 'average' importance of authority and hub scores.

We also highlight other features of μCO-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, μCO-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 nVSM. Third, if λ_x and λ_d are all equal to 1, μCO-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 nVSM. Also, $H(d;t)^1$ is calculated based on $A(u;t)^1$.

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: nVSM (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 nVSM and the two GM approaches [9, 19]. Further, to show the capability of the second knowledge facet of ExpFinder, i.e., μCO-HITS, over topic-sensitive CO-HITS (simply CO-HITS), we compare ExpFinder with an alternative ExpFinder form that combines nVSM and CO-HITS (Section 5.5). Finally, we summarise our evaluation (Section 5.6).

5.1 Datasets

We use four benchmark datasets in our evaluation. One is the Lattes Expertise Retrieval (*LExR*)[2] test collection [17] 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 datasets[3] are Information Retrieval (IR), Semantic Web (SW), and Computational Linguistics (CL) which are filtered subsets of DBLP dataset [18]. 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.

TABLE 1: A summary of our four datasets

	LExR	IR	SW	CL
# of documents	14879	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

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 [38], [9] and [7] used two datasets with seven topics, two datasets with 13 and 203 topics and one dataset with 14 topics, respectively[4]. 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 $\mathcal{D}$ in each dataset. Thus, given each t_g, we find its most similar phrase t from $\mathcal{D}$. Then, t is alternatively used as a topic, instead of t_g. To find t given t_g, we use the scientific pre-trained model SciBERT[5] [39] 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 $\mathcal{D}$. Then, we measure their similarity as:

$$sim(s_1, s_2) \approx \cos(\vec{s_1}, \vec{s_2}) = \frac{\vec{s_1} \cdot \vec{s_2}}{\|\vec{s_1}\| \, \|\vec{s_2}\|}, \qquad (12)$$

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, $s_1 = (w_1, \ldots, w_n)$, then, $\vec{s_1} \approx \frac{1}{n}(\vec{w_1} + \ldots + \vec{w_n})$, where $(\vec{w}_1, \ldots, \vec{w}_n)$ are the embedded vectors of $(w_1, \ldots, w_n)$. 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 nTFIDF in nVSM in calculating $p(t|d)$ in Eq. 3; (2) Thus, we also fix w_{dx}=1 and w_{du}=1 in Eq. 10 and Eq. 11; and (3) from our empirical testing, we observed Eq. 10 and Eq. 11 are commonly converged after 5 iterations, so we set $k = 5$.

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

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

4. These datasets are also no longer publicly available.

5. 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		SW		CL	
Topic	Phrase	Topic	Phrase	Topic	Phrase	Topic	Phrase
synthesis	synthesis	information retrieval	information retrieval	semantic web	semantic web	question answering	question answering
risk factor	risk factor	search engine	search engine	linked data	linked data	knowledge transfer	knowledge transfer
public health	public health	patent search	patent search	text understanding	text understanding	summarization	summarization
thin film	ultrathin film	data modelling	information modelling	auction	bidding	machine vision	computer vision
development validation	validation process	cooperative work	collaborative working	invalidity	inadequacy	image classification	image recognition

For all expert finding models in our evaluation, our aim is to generate a ranked list of experts based on the outcome of the formula: (1) Eq. 11 in ExpFinder, (2) Eq. 10 in CO-HITS, (3) Eq. 3 and Eq. 4 in DLM, (4) Eq. 3 and Eq. 7 in TFIDF-based VSM, and (5) Eq. 3 and Eq. 8 in nVSM, where all these methods are compared, in addition to WISER [8] and RepModel [19].

We use two widely-used evaluation metrics for expert finding [7, 16]: (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 [16]:

$$P@n = |S_n \cap R_t|/n, \tag{13}$$

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 [7]:

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

where i is the rank, $rel(i)$ 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 [38].

5.3 Evaluation of nVSM

As nVSM is one key compoenent 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 Eq. 3 and Eq. 7 (denoted as TFIDF); (2) *The DLM model* [6, 7] denoted using Eq. 3 and Eq. 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 [6] and [7], respectively; and (3) A recent probabilistic model *WISER* [8] 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 [40]: $\sum_{d}^{\mathcal{D}_{x,t}} \frac{1}{rank(d)}$ that represents the ranks of x's documents where t appears ($\mathcal{D}_{x,t}$). Since t is a phrase, $\mathcal{D}_{x,t}$ consists of the subset of $\mathcal{D}_x$ that any of t's constituent terms appears. $rank(d)$ is the ranking position of a document d out of $\mathcal{D}$, where the position is determined by BM25 [24]. The hyper-parameters $k1$ and b in BM25 are set to be 1.2 and 0.75, respectively, based on the suggestion [41].

The evaluation results are presented in Fig. 3 that shows the AP values with n (n=10, 15, . . ., 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 LExR, 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 LExR. 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. Also, 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 μCO-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

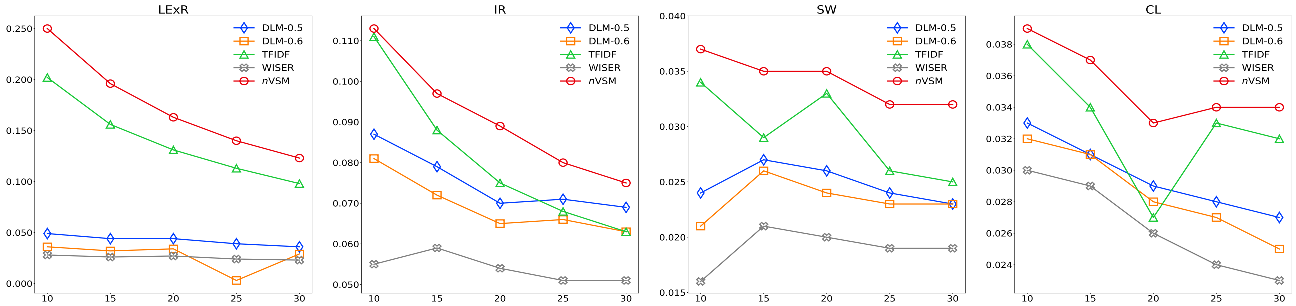


Fig. 3: Comparison on the average precision (AP) values: x-axis shows n of P@n and y-axis shows AP values.

TABLE 3: MAP and the 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%)
nVSM	**0.666**		**0.222**		**0.106**		0.106		**0.275**	

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, . . ., 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:

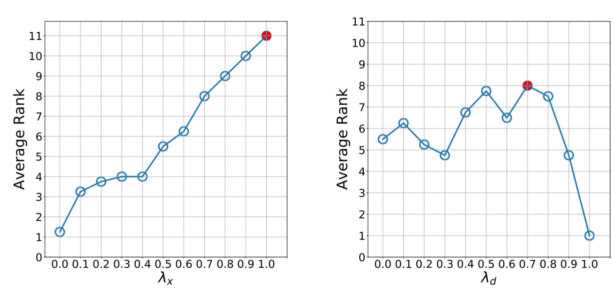
$$Avg(a, \lambda_x) = \frac{1}{|Z|} \sum_{b \in Z} \text{MAP}(a, b). \tag{15}$$

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 $Avg(Z, \lambda_x) = [1, 0.9, 0.8, \ldots, 0]$, then the corresponding element-wise rank vector is $R(Avg(Z, \lambda_x)) = [11, 10, 9, \ldots, 1]$, 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:

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

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.

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: $Avg(b, \lambda_d) = \text{MAP}(a, b)$. 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, λ_x = 1.0 produces the highest rank, whereas λ_d = 0.7 is the highest rank with λ_x = 1.0. The best ones are denoted in red color.



(a) (b)

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

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 two GMs: *ADT* [9] and *Reputation Model* (simply *RepModel*) [19]. Further, we compare μCO-HITS (Eq. 11) with CO-HITS (Eq. 10) to validate the stronger capability of μCO-HITS's variation approach over CO-HITS. Finally, we show that ExpFinder works well regardless of *topic coverage*.

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

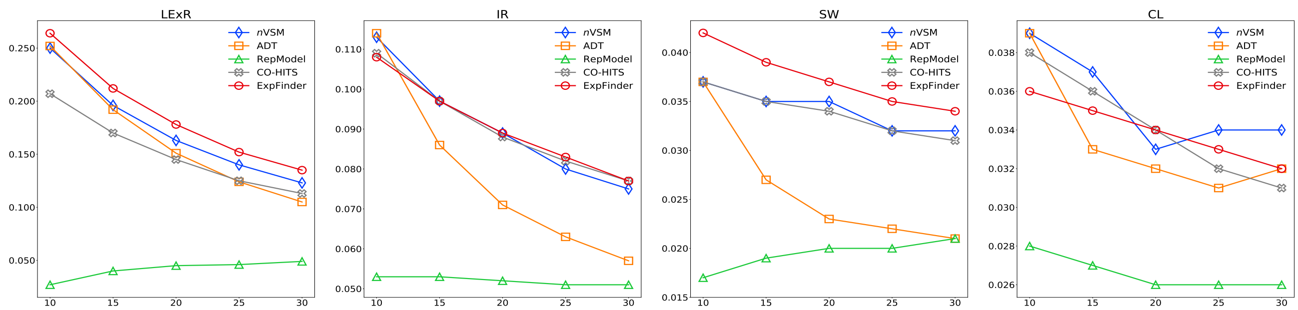


Fig. 5: Comparison on the average precision (AP) values: x-axis shows n of P@n and y-axis shows AP values.

TABLE 4: MAP and the the improvement ratio of ExpFinder.

	LExR		IR		SW		CL		Avg.	
nVSM	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**	

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. ADT ranks x given a topic t based on the score function $s(x,t)$ (the higher the more important):

$$s(x,t) = \sum_{d \in \mathcal{D}_x} w_{xd} \cdot pweight(d,t), \tag{17}$$

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 [19] 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 [19]. 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 $\mathcal{D}_x$, 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 $weight(x,w)$ for all experts, and $weight(x,w)$ is the weight of x on w calculated by the number of documents in $\mathcal{D}_x$ where w appears. We also set k=5 as done for μCO-HITS.

For the fair comparison between μCO-HITS and CO-HITS, as μCO-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. 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 nVSM into CO-HITS.

Fig. 5 shows the evaluation results based on the AP values with n (n=10, 15, ..., 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, ..., 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 μCO-HITS can have a competitive potential for improving the performance over CO-HITS. Third, to determine the impact of incorporating nVSM into μCO-HITS in ExpFinder, let us compare ExpFinder with nVSM. As observed, ExpFinder is clearly and consistently better than nVSM over LExR and SW, although they are similar on IR and nVSM looks better than ExpFinder on CL. On average, we observe that our ensemble model ExpFinder combining nVSM with μCO-HITS is observed to be more powerful than only nVSM. Table 4 shows the evaluation results in MAP. The best one is denoted in boldface. As

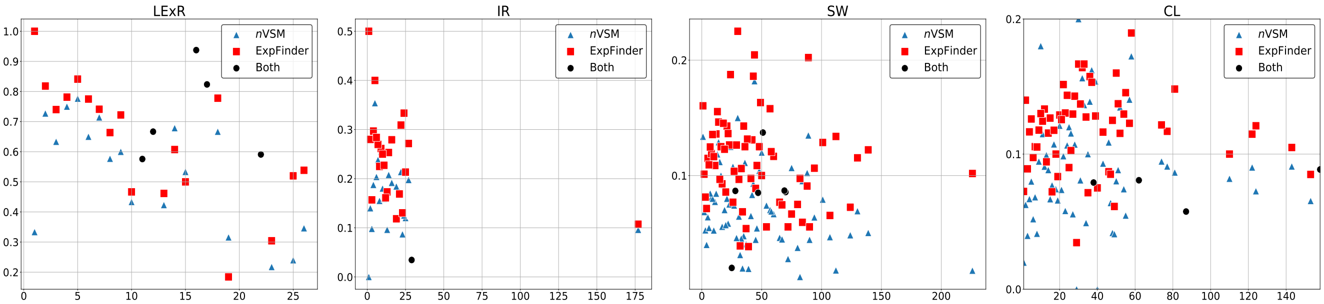


Fig. 6: MAP values of nVSM and ExpFinder: the x-axis shows topic coverage values, and the y-axis shows the MAP values (topic coverage of a topic t means the number of known experts associated with t).

observed, ExpFinder outperforms all the methods in 11.6%, 30.1%, 129.1% and 19.0% over nVSM, ADT, RepModel and CO-HITS, respectively, on average. Interestingly, nVSM is observed as the second better one. This also shows that our nVSM 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 nVSM 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 nVSM and ExpFinder have the same MAP value. We can observe the following: (1) On LExR, ExpFinder dominantly outperforms or is equal to nVSM 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 nVSM 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 prevailingly outperforms nVSM across most of the topic values as ExpFinder's distribution is predominantluy higher than nVSM's distribution; (4) On CL, although nVSM is better than ExpFinder on the topic coverage values 10 and 30, ExpFinder is observed notably better than nVSM 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 nVSM regardless of topic coverage, showing the validity of the design paradigm of ExpFinder, that is, incorporating nVSM into μCO-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 nVSM and μCO-HITS, and compared the results with other expert finding models: TF-IDF based VSM (denoted as TFIDF), DLM-0.5 [6] and DLM-0.6 [7], WISER [8], ADT [9] and RepModel [19].

We showed the capability of nVSM using different AP values (n=10, 15, …, 30) and MAP, in comparison with TFIDF, DLM-0.5 and DLM-0.6. On average, the improvement ratio of nVSM 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 Table 3 and Table 4 in MAP. That is, we showed 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%; nVSM in 11.6%; RepModel in 129.1%; and CO-HITS in 19.0%. Further, we showed that ExpFinder incorporating nVSM into μCO-HITS indeed improves only nVSM. It means that exploiting network propagation effects on ECG using μCO-HITS with the outcome of nVSM can contribute to better estimating topic-sensitive weights of experts. Also, by comparing ExpFinder with a model combining nVSM with CO-HITS, we proved that μCO-HITS can be an effective approach for improving CO-HITS for expert finding. 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 ensemble model ExpFinder for expert finding is effective and competitive for expert finding.

As future work, it could be worth to try to find a way 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 [8] reported that its best MAP values are 0.214 and 0.363 on the two datasets, BMExpert [7] also showed 0.06 as the best MAP value of the DLM [6] on the single dataset, and ADT [9] also showed its best MAP values are 0.0943 and 0.1986 on the two datasets. As another future work, we plan to accommodate a general expertise knowledge source as [8], e.g. Wikipedia, into ExpFinder to see its potential for enhancing ExpFinder's capability. Another interesting future work is that we could 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 vector values.

6 Conclusion

In this paper, we proposed ExpFinder a novel ensemble 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 [17], Information Retrieval, Semantic Web and Computational Linguistics in DBLP dataset [18]) from the academic domains. The novelty of ExpFinder is in its incorporation of a novel N-gram vector space model (nVSM) into μCO-HITS. The key to designing nVSM is to utilise the state-of-the-art IDF method [15] for estimating the topic-sensitive weights of experts given a topic. Such estimated weights are further improved by incorporating them into ECG using μCO-HITS. Our novelty of μCO-HITS is to design two variation schemes of CO-HITS [16], thus proposing a unified formula for successfully integrating nVSM with μCO-HITS. We showed comprehensive evaluation, comparing ExpFinder with the six representative models, that is, TF-IDF based vector space models (TFIDF), two document language models (DLM) [6, 8], two graph-based models (GM) [9, 19], and topic-sensitive CO-HITS [16]. We showed ExpFinder is a highly effective ensemble model for expert finding, outperforming TFIDF with 35.9%, DLMs with 71.5% - 160.2%, GMs with 31% - 129.1%, and topic-sensitive CO-HITS with 19.0%.

References

[1] O. Husain, N. Salim, R. A. Alias, S. Abdelsalam, and A. Hassan, "Expert finding systems: A systematic review," *Applied Sciences*, vol. 9, no. 20, p. 4250, 2019.

[2] M. Stankovic, C. Wagner, J. Jovanovic, and P. Laublet, "Looking for experts? what can linked data do for you?" in *Proceedings of the WWW2010 Workshop on Linked Data on the Web*, 2010.

[3] R. Gonçalves and C. F. Dorneles, "Automated expertise retrieval: A taxonomy-based survey and open issues," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–30, 2019.

[4] C. T. Chuang, K. H. Yang, Y. L. Lin, and J. H. Wang, "Combining query terms extension and weight correlative for expert finding," in *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, vol. 1. IEEE, 2014, pp. 323–326.

[5] O. Alhabashneh, R. Iqbal, F. Doctor, and A. James, "Fuzzy rule based profiling approach for enterprise information seeking and retrieval," *Information Sciences*, vol. 394, pp. 18–37, 2017.

[6] K. Balog, L. Azzopardi, and M. de Rijke, "A language modeling framework for expert finding," *Information Processing & Management*, vol. 45, no. 1, pp. 1 – 19, 2009.

[7] B. Wang, X. Chen, H. Mamitsuka, and S. Zhu, "Bmexpert Mining medline for finding experts in biomedical domains based on language model," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 12, no. 6, pp. 1286–1294, Nov. 2015.

[8] P. Cifariello, P. Ferragina, and M. Ponza, "WISER: A semantic approach for expert finding in academia based on entity linking," *Information Systems*, vol. 82, pp. 1 – 16, 2019.

[9] S. D. Gollapalli, P. Mitra, and C. L. Giles, "Ranking experts using author-document-topic graphs," in *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*, ser. JCDL '13, 2013, pp. 87–96.

[10] C. S. Campbell, P. P. Maglio, A. Cozzi, and B. Dom, "Expertise identification using email communications," in *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, ser. CIKM '03, 2003, p. 528–531.

[11] R. Yeniterzi and J. Callan, "Constructing effective and efficient topic-specific authority networks for expert finding in social media," 2014, p. 45–50.

[12] M. S. Faisal, A. Daud, A. U. Akram, R. A. Abbasi, N. R. Aljohani, and I. Mehmood, "Expert ranking techniques for online rated forums," *Computers in Human Behavior*, vol. 100, pp. 168–176, 2019.

[13] K. Bok, I. Jeon, J. Lim, and J. Yoo, "Expert finding considering dynamic profiles and trust in social networks," *Electronics*, vol. 8, no. 10, p. 1165, 2019.

[14] B. Sziklai, "How to identify experts in a community?" *International Journal of Game Theory*, vol. 47, no. 1, pp. 155–173, 2018.

[15] M. Shirakawa, T. Hara, and S. Nishio, "IDF for Word N-Grams," *ACM Trans. Inf. Syst.*, vol. 36, no. 1, Jun. 2017.

[16] H. Deng, M. R. Lyu, and I. King, "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*, 2009, p. 239–248.

[17] V. Mangaravite, R. L. Santos, I. S. Ribeiro, M. A. Gonçalves, and A. H. Laender, "The LExR Collection for Expertise Retrieval in Academia," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2016, p. 721–724.

[18] G. Bordea, T. Bogers, and P. Buitelaar, "Benchmarking domain-specific expert search using workshop program committees," in *Proceedings of the 2013 workshop on Computational scientometrics: theory & applications*, 2013, pp. 19–24.

[19] D. Schall, *A Social Network-Based Recommender Systems*. Springer, 2015.

[20] F. Riahi, Z. Zolaktaf, M. Shafiei, and E. Milios, "Finding expert users in community question answering," in *Proceedings of the 21st International Conference on World Wide Web*, 2012, pp. 791–798.

[21] L. K. Schunk and G. Cong, "Using transactional data from erp systems for expert finding," in *Database and Expert Systems Applications*, 2010, pp. 267–276.

[22] V. Mangaravite and R. L. Santos, "On information-theoretic document-person associations for expert search in academia," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '16. Association for Computing Machinery, 2016, p. 925–928.

[23] C. Van Gysel, M. de Rijke, and M. Worring, "Unsupervised, efficient and semantic expertise retrieval," in *Proceedings of the 25th International Conference on World Wide Web*, 2016, pp. 1069–1079.

[24] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: Bm25 and beyond," *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[25] X. Jiang, X. Sun, Z. Yang, H. Zhuge, and J. Yao, "Exploiting heterogeneous scientific literature networks to combat ranking bias: Evidence from the computational linguistics area," *Journal of the Association for Information Science and Technology*, vol. 67, no. 7, pp. 1679–1702, 2016.

[26] C. L. Koumenides and N. R. Shadbolt, "Ranking methods for entity-oriented semantic web search," *Journal of the Association for Information Science and Technology*, vol. 65, no. 6, pp. 1091–1106, 2014.

[27] J. Zhang, M. S. Ackerman, and L. Adamic, "Expertise networks in online communities: structure and algorithms," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 221–230.

[28] G. A. Wang, J. Jiao, A. S. Abrahams, W. Fan, and Z. Zhang, "Expertrank: A topic-aware expert finding algorithm for online knowledge communities," *Decision support systems*, vol. 54, no. 3, pp. 1442–1451, 2013.

[29] D. Kundu and D. P. Mandal, "Formulation of a hybrid expertise retrieval system in community question answering services," *Applied Intelligence*, vol. 49, no. 2, pp. 463–477, 2019.

[30] H. Deng, I. King, and M. R. Lyu, "Enhanced models for expertise retrieval using community-aware strategies," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 1, pp. 93–106, 2011.

[31] D.-R. Liu, Y.-H. Chen, W.-C. Kao, and H.-W. Wang, "Integrating expert profile, reputation and link analysis for expert finding in question-answering websites," *Information processing & management*, vol. 49, no. 1, pp. 312–329, 2013.

[32] c. Aslay, N. O'Hare, L. M. Aiello, and A. Jaimes, "Competition-based networks for expert finding," in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '13, 2013, p. 1033–1036.

[33] N. Torkzadeh Mahani, M. Dehghani, M. S. Mirian, A. Shakery, and K. Taheri, "Expert finding by the dempster-shafer theory for evidence combination," *Expert Systems*, vol. 35, no. 1, 2018.

[34] Y.-B. Kang, P. Delir Haghighi, and F. Burstein, "Cfinder: An intelligent key concept finder from text for ontology development," *Expert Syst. Appl.*, vol. 41, no. 9, p. 4494–4504, 2014.

[35] T. Roelleke and J. Wang, "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*, 2008, p. 435–442.

[36] K. Lu, "An insight into vector space modeling and language modeling," in *iConference*, 2013.

[37] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, p. 604–632, Sep. 1999.

[38] H. Deng, I. King, and M. R. Lyu, "Formal Models for Expert Finding on DBLP Bibliography Data," in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ser. ICDM '08, 2008, pp. 163–172.

[39] I. Beltagy, K. Lo, and A. Cohan, "Scibert: Pretrained language model for scientific text," in *EMNLP*, 2019.

[40] C. Macdonald and I. Ounis, "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*, ser. CIKM '06, 2006, p. 387–396.

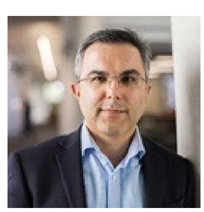
[41] Y. Lv and C. Zhai, "Adaptive term frequency normalization for bm25," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, ser. CIKM '11. Association for Computing Machinery, 2011, p. 1985–1988.

Yong-Bin Kang received his PhD in Information Technology from Monash University in 2011. Currently, he is a senior data science research fellow in the fields of natural language processing, machine learning and data mining at Swinburne University of Technology. He has been working on large industrial, multi-disciplinary research projects such as patent analytics, clinical data analytics, scientific-article analytics, social-media data analytics, expert finding and matching, and machine learning algorithms and applications.

Hung Du is currently studying his final year Bachelor degree in Computer Science at Swinburne University of Technology, Australia and working as research assistant. His research interest include Natural Language Processing, Reinforcement Learning, Data Mining and Applied Machine Learning.

Abdur Rahim Mohammad Forkan received his PhD degree in Computer Science from the RMIT University, Australia in 2016. He is a Senior Research Engineer at the Digital Innovation Lab, Swinburne University of Technology, Australia and working on several multi-disciplinary industrial research projects. His research interest include Data Analytics, Health Informatics, Pervasive Systems, Cloud Computing, Data Mining and Applied Machine Learning.

Prem Prakash Jayaraman is an associate Professor and the Head of Digital Innovation Lab at Swinburne. He is broadly interested in the research areas of Internet of Things, Mobile and Cloud Computing, Health Informatics and application of Data Science techniques and methodologies in real-world settings. He was a key contributor and one of the architects of the Open Source Internet of Things project (OpenIoT) that has won the prestigious Black Duck Rookie of the Year Award in 2013 (https://github.com/OpenIotOrg/openiot).



Amir Aryani is the Head of the Social Data Analytics (SoDA) Lab at the Swinburne University of Technology. He has experience with large-scale and cross-institution projects in Australia and Europe. His track records include collaboration with high-profile international institutions such as British Library, ORCID, Data Archiving and Network Analysis (DANS, Netherlands), and funders including ARC, NHMRC, and NIH. He has published high impact journals such as Nature Scientific Data, Metadata and Semantics Research, and Frontiers in Artificial Intelligence and Applications.

Timos Sellis received the PhD degree in Computer Science from the University of California, Berkeley, in 1986. He is now a visiting scientist at Facebook and also adjunct professor at Swinburne. Between 2013 and 2015, he was a professor at RMIT University, Australia, and before 2013, the director of the Institute for the Management of Information Systems and a professor at the National Technical University of Athens, Greece. He is a fellow of the IEEE and ACM. In 2018, he was awarded the IEEE TCDE Impact Award, in recognition of his impact in the field and for contributions to database systems and data engineering research.