Learning to Find Topic Experts in Twitter via Different Relations

Wei Wei, Gao Cong, Chunyan Miao, Feida Zhu and Guohui Li

Abstract—Expert finding has become a hot topic along with the flourishing of social networks, such as micro-blogging services like Twitter. Finding experts in Twitter is an important problem because tweets from experts are valuable sources that carry rich information (e.g., trends) in various domains. However, previous methods cannot be directly applied to Twitter expert finding problem. Recently, several attempts use the relations among users and Twitter Lists for expert finding. Nevertheless, these approaches only partially utilize such relations. To this end, we develop a probabilistic method to jointly exploit three types of relations (i.e., follower relation, user-list relation and list-list relation) for finding experts. Specifically, we propose a Semi-Supervised Graph-based Ranking approach (SSGR) to offline calculate the global authority of users. In SSGR, we employ a normalized Laplacian regularization term to jointly explore the three relations, which is subject to the supervised information derived from Twitter crowds. We then online compute the local relevance between users and the given query. By leveraging the global authority and local relevance of users, we rank all of users and find top-N users with highest ranking scores. Experiments on real-world data demonstrate the effectiveness of our proposed approach for topic-specific expert finding in Twitter.

Index Terms—Expert search, micro-blogging, twitter, list, graph-based ranking.

1 INTRODUCTION

Expert finding (a.k.a., expert search [9]), which aims at identifying people with the relevant expertise or experiences on a given topic query, has been studied broadly in domains such as enterprise [10], [9], question answering [11], [12], [42], [44], Web [13] and academic society [14], [15].

Recently, expert finding problem has gained increasing attention in social media [16], [17], such as micro-blogging services like Twitter, a new type of social media in providing a publicly available channel for users to publish 140-character short messages (i.e., tweets). Twitter has gained huge popularity and gathered a tremendous amount of tweets in recent years. These tweets cover extremely wide and diverse topics, such as routine activities or experiences, top news, technology, and myriad of other highly specialized areas, etc. Correspondingly, users in Twitter have rich expertise on various topics and finding these topic-specific experts paves a way to enable others to retrieve or follow the relevant and trustworthy information on a specific topic in micro-blogging services [3], [4], [5]. For example, if a Twitter user wants to follow expert users for receiving tweets that are highly relevant for an event topic like “Boston Marathon bombings”, or follow users whose tweets are worthy of reading for a domain-specific topic like “machine learning”. In addition, identifying such users is also a preprocessing step towards many applications like opinion mining [6] and name entity recognition (NER) [8], [3], [7]. For instance, opinions mined from beauticians’ tweets are more likely to favor a cosmetic manufacturer (e.g., Dior) than those from common users.

Nevertheless, the problem of Twitter expert search differs from the conventional expert search problem [14], [15], [13], [10], [9]. They generally rely on the assumption that all the documents associated with the candidate experts contain tacit knowledge related to the expertise of individuals [10], [9], whereas it might not be true in Twitter, as users’ published tweets might not be directly related to their expertise, such as a rumormonger [1], [2], who is not an expert, but may publish/retweet a substantial amount of tweets containing the topic words. Therefore, the problem of expert finding in Twitter is more challenging.

Indeed, there exist several attempts for the Twitter expert finding problem. For example: (i) Several traditional methods like PageRank-based method [3] and clustering-based method [4], make use of the follower relations as well as users’ bios and tweets to infer the general influence of users on different topics; and (ii) A recent study (Cognos [5]) proposes to identify topic-specific experts by mining the meta-data of Twitter Lists. A Twitter list is usually created by a user to group her followings according to a criterion, e.g., having expertise on “data mining”. Intuitively, the meta-data (e.g., title) of a list can be viewed as
Fig. 1: Example. An illustration of different types of relations among users and lists.

the crowdsourced topical annotations of users in that list [5]. For instance, a user involved in a list named “machine learning” is likely to have expertise on machine learning. Hence, a user contained in many lists on a theme is very likely to be an expert on that topic [5]. It has been found that Cognos [5] utilizing user-list relation is more effective for inferring the expertise of users than previous methods based on follower relations as well as users’ bios and tweets.

However, existing approaches [3], [4], [5] only partially utilize either follower relation or user-list relation alone, and they are thus insufficient for Twitter expert finding problem. To this end, we propose to jointly exploit such relations for accurately inferring the users’ domain of expertise. We illustrate this with an example in Fig. 1. Without loss of generality, suppose we infer the expertise of user C only depending on the user-list relation between user C and List1 (“Computer Science, CS”), and thus we only deduce that user C has expertise on CS. However, if we exploit more available relations (e.g., follower relation or list-list relation), we can more accurately identify the expertise of user C, such as: (1) The majority of users (e.g., user A, B and D) in List2 (“Natural Language Processing, NLP”) are the followers of user C. Hence, user C is likely to be an expert on NLP. Furthermore, the semantic similarity between List1 and List2 can be used to strengthen this possibility. (2) Similarly, we can infer that user C may have expertise on ML (“Machine Learning”). Consequently, by means of more relations, we not only certify that user C is more likely an expert on CS, but also refine her possible specific expertise on CS, i.e., NLP and ML.

Therefore, we propose an approach to jointly exploit the different types of relations among users and lists for improving the accuracy of finding experts on a given topic in Twitter. Specifically, we take into account two types of information to target Twitter expert finding problem, namely: (i) Local Relevance, the similarity between users’ published tweets and the given query; and (ii) Global Authority, the global expertise scores of users on a given topic in Twitter.

Consequently, we estimate the probability of each user being an expert on a given topic from two aspects. First, we propose an innovative Semi-Supervised Graph-based Ranking approach, called SSGR, to compute the global authority of users on a given topic, by jointly exploiting different types of relations in Twitter Lists and follower graphs. To the best of our knowledge, this is the first attempt that targets expert finding problem in Twitter by utilizing all of these relations. In particular, SSGR employs: (a) a normalized Laplacian regularization term to smooth the ranking of users and lists on three different topic-specific graphs; and (b) a loss term to ensure the global authority of users are in accordance with the wisdom of Twitter crowds. Second, we also propose a Gaussian-based method to estimate the local relevance of candidates for arbitrary topical queries. The extensive experiments conducted on a real-world data set demonstrate the effectiveness of our proposed method over the state-of-the-art baselines.

Roadmap. The remaining of the paper is organized as follows. In Section 2, we review the related work. The proposed method is presented in Section 3, Section 4 and Section 5, followed by the experimental results in Section 6. Finally, Section 7 concludes this paper.

2 RELATED WORK

Early Work: Expert finding problem is initially proposed in knowledge management community [19], which primarily builds a repository by manually creating the skill descriptions of experts. However, the manual process is time consuming and expensive. Hence, many automatic approaches are proposed for the expert finding problem [21]. The task of expert finding has attracted extensive attention of information retrieval community since it was included in TREC enterprise track [20]. Most of work [10], [22], [9], [18] on organization expert search problem generally falls into two categories, namely profile-centric and document-centric methods. These methods assume that individuals’ published documents are relevant to their expertise with different degrees of match, and they focus on modeling the associations between documents and candidate experts. In contrast, tweets are clearly attached to their publishers. Additionally, there exist many methods of expert finding within other domains, such as academic search [14], [15], Web [13] and so on.

Influential User Identification: Most of existing work on expert identification in social networks focuses on finding influential users from different types of social networks, such as Community Question Answering [12], [11], [42], Blog [27], Academic Social Network [23], Twitter [33], [43], and other social networks [38], [41]. However, these methods do not consider topical dimension, which thus cannot identify topic-specific experts as we do in this work.

Topic-specific Expert Finding: Several proposals [5], [4], [3] approach the problem of identifying topic-
specific experts in Twitter. Weng et al. [3] propose an approach called TwitterRank, which works in two steps. First, it employs Latent Dirichlet Allocation (LDA) model [30] to detect the topics of individuals based on their tweets. Second, for each topic, it builds a weighted graph by considering both the topical similarity between two users and follower graph, and then employ PageRank algorithm [34] to find topic-specific influential users. TwitterRank tends to select the well-known users that typically are highly visible over the follower graph. Another approach proposed by Fal et al. [4], extracts users’ features from the follower graph and users’ posted tweets, and then employs a Gaussian-based mixture model to cluster users for ranking. One fundamental difference of our approach from TwitterRank and Pal’s work is that we utilize the wisdom of Twitter crowds, enclosed in Twitter Lists, as the supervised information to infer the topical expertise of users. Correspondingly, our approach handles the user-user relation, user-list relation and list-list relation while TwitterRank and Pal’s work only consider user-user relation.

Ghosh et al. [37] propose to utilize Twitter List to analyze the attributes of Twitter users. In their subsequent work, they develop a system named Cognos [5] to infer the topical expertise of users by utilizing only user-list relation in Twitter Lists, which captures the wisdom from Twitter crowds. Cognos represents each user by the meta-data of Twitter lists that contain the user, and then employs a similarity measure [31] to compute the similarity score between each user and a topical query, which is used to rank users for search. Intuitively, Cognos tends to choose users that are contained in many lists whose meta-data contain the query. The experimental results show that Cognos outperforms the approach using social relations [4]. In contrast, our method is able to make use of three types of relations for more accurately identifying experts.

Graph-based Ranking: There exist several attempts to model the process of expert finding by graph-based ranking methods [26], [24], such as Hyperlink-Induced Topic Search (HITS) based expert authority [25], PageRank-based user influence [3], and probabilistic random walk on expertise graphs [18]. However, these methods heavily rely on a single type of relations and are usually topic-irrelevant. In contrary, we take into account three different types of relations to identify the topic-specific experts in Twitter.

Other work on Twitter List: In addition, Twitter lists have also been used for other purposes such as entity link (ZenCrowd) [32] and news curators finding [33]. Welch et al. [40] utilize the Lists features as a context to find the source of topic information. However, the purposes of these studies are different from the current work and thus will not be discussed in detail.

3 Overview of Proposed Approach

We first give the statement of our expert search problem, in Section 3.1, and then present an overview of our proposed approach.

3.1 Problem Statement

Let the set of Twitter users be $\mathcal{U} = \{u_i\}_n$ (n is the number of users), which are candidate experts. User $u_i$’s posted tweets, bio and the meta-data of lists containing $u_i$ are concatenated and form a pseudo-document, which is referred to as the context of $u_i$, denoted by $d^n_i$.

A topic query is defined as $Q$ comprising several terms, namely $Q = \{t_1, \ldots, t_{|Q|}\}$, where $|Q|$ is the number of terms. Then, the topic-specific expert finding problem is to rank a set of candidate experts $\mathcal{U}$ based on the relevance of their expertise to the topic query $Q$. The probability of user $u_i$ in $\mathcal{U}$ being an expert on $Q$ can be estimated via Bayes theorem by following previous work on expert search:

$$P(u_i|Q) = \frac{P(Q|u_i)P(u_i)}{P(Q)} \propto P(Q|u_i)P(u_i),$$

where $P(Q)$ is the prior probability of $Q$; $P(u_i)$ is the prior probability of candidate $u_i$. As $P(Q)$ is the same for all candidate experts and $P(u_i)$ is generally assumed uniform over $\mathcal{U}$ [2], they do not affect the rankings of candidate experts, and thus are ignored.

Therefore, the problem is transformed to estimate the probability of a query $Q$ given candidate $u_i$, i.e., $P(Q|u_i)$. Many language models are proposed for this task [10], [9], [18]. By following the work [3], we treat each term $t$ in $Q$ as a potential topic, and adopt the query likelihood model to approximately estimate the probability $P(Q|u_i)$,

$$P(Q|u_i) = \prod_{t_j \in Q} P(t_j|u_i),$$

and

$$L(Q, u_i) = \sum_{t_j \in Q} \log(P(t_j|u_i)) \propto \sum_{t_j \in Q} \log(P(u_i|t_j)P(t_j)),$$

where $L(Q, u_i) \equiv \log(P(Q|u_i))$; $P(u_i|t_j)$ indicates the probability of candidate $u_i$ being an expert on $t_j$ over $\mathcal{U}$; $P(t_j)$ is the prior probability of $t_j$. Similarly, it is uniform for all candidates and thus is ignored.

An expert to query $Q$ should not only be the authority on $Q$, but also publish many relevant tweets containing the terms of $Q$. To characterize the two aspects, we incorporate two factors in estimating $L(Q, u_i)$: (i) global authority; and (ii) local relevance.

Global Authority. It indicates the global expertise score of a user $u_i$ on a potential topic $t_j$ in Twitter, i.e., $P(u_i|t_j)$. By following the work [18], it can be calculated as follows,

$$P(u_i|t_j) = \frac{H(u_i,t_j)}{\sum_{u' \in \mathcal{U}} H(u',t_j)},$$

where $H(u_i,t_j)$ is the scoring function that assigns a score to user $u_i \in \mathcal{U}$ proportional to $u_i$’s global authority on $t_j$. We will present our proposed method of computing the global authority in Section 4.
Fig. 2: Overview of proposed approach.

Local Relevance. It denotes the local similarity between user $u_i$ and the given query $Q$ over the context $d^{u_i}$ of $u_i$. To consider the sequence of terms in $Q$, we take each two adjacent terms in $Q$ for computing the local relevance, $K(t_j, t_{j+1}; d^{u_i})$. For example, given query $Q = \{t_1, t_2, t_3\}$, we compute $K(t_1, t_2; d^{u_i})$ and $K(t_2, t_3; d^{u_i})$.

Consequently, Eq. (2) is converted as follows,

$$L(Q, u_i) \propto \sum_{t_j \in Q} \log \left( \Pr(u_i | t_j) \right)$$

$$\propto \frac{1}{2} \sum_{j=1}^{|Q|-1} \left( \log(\Pr(u_i | t_j)) + \log(\Pr(u_i | t_{j+1})) \right) K(t_j, t_{j+1}; d^{u_i})$$

$$\propto \frac{1}{2} \sum_{j=1}^{|Q|-1} \left( \Pr(u_i | t_j) \Pr(u_i | t_{j+1}) \right) K(t_j, t_{j+1}; d^{u_i})$$

In particular, when $|Q| = 1$, $L(Q, u_i) \propto \log \left( \Pr(u_i | t_1) \right)$.

The remaining problem is how to compute the global authority score $A(u_i, t_j)$ and local relevance $K(t_j, t_{j+1}; d^{u_i})$. We will detail them in the following sections, respectively.

3.2 Overview

Next, we present an overview (as shown in Fig. 2) of our approach to addressing the topic-specific expert finding problem. Specifically, it consists of two components, namely, an offline graph-based ranking algorithm (called SSGR, detailed in Section 4) to learn the global authority of each candidate and an online ranking model (named RM, detailed in Section 5.2) to select top-$N$ relevant experts on the given query. In particular, each term $t$ in Twitter is treated as a potential topic by following the work [3].

- We first construct an authority matrix (similar to the inverted index) $R$ over the Twitter corpus. Specifically, each row $R_i \in R$ is offline computed by SSGR for each term $t$ in Twitter, in which we jointly exploit the three different relations of users and Twitter lists for inferring the global authority of each candidate on $t$ in Twitter.
- For a given topic query $Q = \{t_1, \cdots, t_Q\}$, we use an online ranking model (i.e., RM), based on the corresponding rows in $R$ for terms contained in $Q$, to select top-$N$ users as experts on $Q$, by taking into account the global authority and local relevance of candidates (cf. Eq. (4)).

2. We removed non-English characters, stopwords, punctuation as well as the high-frequency words in Twitter (e.g., RT), and keep the left.

4 Learning The Global Authority

To learn the global authority of candidate users on a single term topic query (denoted by $Q_t$), we present a novel semi-supervised graph-based ranking method, called SSGR. It is capable of exploiting the different relations (i.e., follower relation, user-list relation and list-list relation) among users and lists to mutually reinforce the ranking of users and lists for inferring the global expertise scores of users on $Q_t$. We present the User-List Interaction (ULI) graph to model the different relations in Section 4.1, followed by the intuitive benefits of jointly exploiting the three relations for calculating the global authority of candidates on a given topic in Section 4.2. Then we present the proposed method SSGR in Section 4.3.

4.1 User-List Interaction Graph (ULI)

In this section, we present the definition of ULI graph (as shown in Fig. 3). For clarity, some notations and their definitions are listed in Table 1.

Let $G = \{U \cup \mathcal{L}, E\}$ be the ULI graph, where $U = \{u_i\}_{n}$ and $\mathcal{L} = \{L_i\}_{m}$ denote a set of $n$ users and a set of $m$ lists, respectively, and $E$ denotes the edge set which comprises three different relations, namely (i) follower relation, an edge between a user and another follower (denoted by $e_{u_i}$); (ii) user-list relation, an edge between a user and a list, which consists of two types of edge: a) MEM-OF relation: an edge between a member user and her included list (denoted by $e_{u_i}$); and b) SUB-TO relation: an edge between a subscribe user and her subscribing list (denoted by $e_{u_i}$); (iii) list-list relation, an edge between two lists (denoted by $e_{u_i}$). Correspondingly, there are three types of topic-specific graphs related to a given topic ($Q_t$), which are:

- $W_u$: a $n$-by-$n$ symmetric topic-specific follower graph, in which $w_{u_i}$ denotes the similarity between user $u_i$ and her follower $u_j$ for a given topic. Note each entry in $W_u$ only considers the symmetric relation between two users, i.e., $u_i$ and $u_j$ follow each other.

3. The topic query corresponds to a single term in Twitter.
**TABLE 1: Notations and definitions.**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query $Q_i$</td>
<td>Given topic query, $Q_i = {w_1, \ldots, w_{</td>
</tr>
<tr>
<td>Node Set $U_L$</td>
<td>User set, $U_L = {u_i}_{i=1}^n$, $n$ is the number of users</td>
</tr>
<tr>
<td>List set $L_i$</td>
<td>List set, $L_i = {L_i^1, \ldots, L_i^m}$, $m$ is the number of lists</td>
</tr>
<tr>
<td>Context $d_{it}$</td>
<td>Context of $u_i$, including $u_i$’s bio, posted tweets and the meta-data of lists containing $u_i$</td>
</tr>
<tr>
<td>$d_{it}^*$</td>
<td>Meta-data of $L_i$</td>
</tr>
</tbody>
</table>

**Graph**

- $W_u$: a $m$-by-$m$ symmetric topic-specific list graph, which is generated based on the mutual $k$-nearest neighbor graph [39], in which $w_{ij}^u$ denotes the similarity between user $u_i$ and her follower $u_j$.
- $W_l$: a $n$-by-$m$ topic-specific user-list graph, in which $w_{il}^l$ denotes the similarity between user $u_i$ and list $L_j$, containing $u_i$ for a given topic. Each entry in $W_l$ refers to MEM-OF relation, i.e., user $u_i$ is included in list $L_j$.

**Similarity Measure**

Given a ULI, one way to compute the similarity $w_{ij}$ (e.g., $w_{ij}^u$, $w_{ij}^l$, or $w_{ij}^m$) between two objects, denoted by $d_i$ and $d_j$ (e.g., user $u$ or list $L$) under $Q_i$, is given in Eq. (5).

$$w_{ij} = \frac{N(Q_i, d_i, d_j)}{N(d_i, Q_i) + N(d_j, Q_i)}$$

where $N(Q_i, d_i, d_j)$ is the co-occurrences of $Q_i$ in $d_i$ and $d_j$, and $N(Q_i, d_i)$ is the number of occurrences of $Q_i$ in $d_i$. However, Eq. (5) might be problematic in some cases, for example, for a topic query like “travel”, assume that user $u_i$ and user $u_j$ have a high overlap between their published tweets on “travel”, but if $u_i$ never mentions the term “travel” in her tweets, $N(Q_i, d_i, d_j)$ will be 0. Moreover, Eq. (5) also ignores the similarity between $u_i$ and $u_j$. To address the problem, we compute the similarity by

$$w_{ij} = \frac{\Pr(Q_i|d_i) + \Pr(Q_i|d_j)}{2} \cos(d_i, d_j),$$

where $\cos(d_i, d_j)$ is the cosine similarity of two objects $d_i$ and $d_j$, each word probability vector (i.e., unigram) of document $d_i$ (or $d_j$) is based on the TF-IDF [36] method, and $\Pr(Q_i|d_i) = \frac{N(Q_i, d_i)}{\sum_{d_i \in D(Q_i)} N(Q_i, d_i)}$.

### 4.2 Intuitions

Recall that the information on users alone might be insufficient for measuring the global authority of candidates on the given topic. We propose to jointly exploit three different types of relations (i.e., follower relation, list-list relation and user-list relation) for inferring the global authority of candidates on $Q_i$. The motivation is based on the intuitions as follows.

- **Intuition 1 (Follower Relation).** Users that are socially connected are more likely to share similar interests (Homophily [3], [28]). Hence (a) if a user is followed by another user with high global authority on $Q_i$, this user is more likely an expert on $Q_i$; and (b) the more followers of a user are experts on a topic, the more likely that the user is an expert on that topic;

- **Intuition 2 (User-list Relation).** In-depth analysis of user-list relation is helpful to infer the expertise of users [5]. We explore two types of user-list relations: (a) MEM-OF relation, i.e., a set of users are included in a list. A list is built by a user to group her followings sharing a common characteristic. Hence, intuitively, i) if a user is relevant to the lists containing her, the user is likely to be an expert on topic $Q_i$ that is relevant to the lists; ii) if a user is contained in many lists relevant to $Q_i$, the user is likely to be an expert on $Q_i$. (b) SUB-TO relation, i.e., a set of users subscribe to a list. It is analogous to follower relation, i.e., this relation is a strong indicator that the users subscribing to a list are interested in the topic of that list. Intuitively, the more subscribers are experts on a topic, the more likely the subscribed list is relevant to that topic. In particular, we use (a)-ii) and (b) of user-list relation as the supervised information in our proposed model.

- **Intuition 3 (List-List Relation):** If a list $L_i$ is highly similar (i.e., $w_{ij}^l$) to another list $L_j$ that is relevant to $Q_i$, list $L_j$ is also likely to be relevant to $Q_i$. Note that exploring the similarity between lists aims to find relevant lists for $Q_i$, which can be used to enhance the relevance of users in such lists to topic $Q_i$.

### 4.3 Semi-supervised Graph-based Ranking

Based on these intuitions, we propose a semi-supervised graph-based ranking method, named SSGR, for computing the global authority of a user on the given topic $Q_i$.

#### 4.3.1 Graph-based Regularization Framework

In this subsection, we present the proposed graph-based regularization framework, which comprises two terms: (i) regularization term, which is used to smooth the ranking scores on the graph; (ii) loss term, which aims to ensure the ranking scores are consistent with the supervision information.

Let $n$-dimensional vector $f = [f_1, \ldots, f_n]^T$ be the ranking scores of users and $m$-dimensional vector $g = [g_1, \ldots, g_m]^T$ be the ranking scores of lists. In particular, the $i$-th entry of $f$ (i.e., $f_i$) denotes the global authority of user $u_i$ on the given topic, and the $i$-th score in $g$ (i.e., $g_i$) denotes the relevance between list $L_i$ and the given topic. In fact, we are only interested in the ranking scores of users to identify topic-specific experts. We also consider the ranking scores for lists in our framework because the ranking scores
of users and lists will reinforce each other mutually as explained in the intuitions in Section 4.2. Formally, the ranking framework is formulated as the following optimization problem:

\[
(f, g) = \arg \min_{f \geq 0, g \geq 0} \left( \mathcal{F}(W_u, W_l, W_{ml}; f, g) + (1 - \lambda)\ell(f, g) \right),
\]

where \(\mathcal{F}\) is a regularization term to smooth the expertise scores (i.e., global authority) of users; the affinity matrices (i.e., \(W_u, W_l, W_{ml}\)) are computed in a topic-specific manner; \(\ell\) is a loss term that aims to ensure the expertise scores of users are consistent with the wisdom of Twitter crowds; and \(\lambda\) is a parameter to trade-off the contributions of regularization term \(\mathcal{F}\) and loss term \(\ell\).

### 4.3.2 Regularization Term

Within the framework, the regularization term aims to give similar ranking scores for similar users (and similar lists) by considering three different types of similarities of users and lists, namely, the similarity between a user and her followers, the similarity between a user and the lists containing her, and the similarity between two lists. The regularization term \(\mathcal{F}\) is defined as follows, based on the principle of normalized Laplacian regularization,

\[
\mathcal{F}(W_u, W_l, W_{ml}; f, g) = \alpha_1 \sum_{i,j=1}^{\text{users}} \left( \frac{f_i}{\sqrt{D_u(i,i)}} - \frac{f_j}{\sqrt{D_u(j,j)}} \right)^2 + \alpha_2 \sum_{i,j=1}^{\text{lists}} \left( \frac{g_i}{\sqrt{D_l(i,i)}} - \frac{g_j}{\sqrt{D_l(j,j)}} \right)^2 + \alpha_3 \sum_{i=1}^{\text{users}} \sum_{j=1}^{\text{lists}} \left( \frac{1}{\sqrt{D_{ul}(i,j)}} - \frac{1}{\sqrt{D_{ul}(j,i)}} \right)^2,
\]

where \(D_u\) and \(D_{ml}\) are \(n\)-by-\(n\) diagonal matrices, \(D_l\) and \(D_{ml}\) are \(m\)-by-\(m\) diagonal matrices. The \((i, j)\)-element of \(D_u\) and \(D_{ml}\) is equal to the sum of \(i\)-th row of \(W_u\), and the sum of \(i\)-th row of \(W_{ml}\). The sum of \(i\)-th column of \(W_{ml}\), respectively. In addition, \(\alpha_1 (\alpha_1 \geq 0\) and \(\alpha_1 + \alpha_2 + \alpha_3 = 1\) is the fusing weight.

Next, we illustrate the first term in the right-hand side of Eq. (8). Minimizing the first term aims to ensure that a user \(u_i\) and her follower \(u_j\) should be assigned similar normalized scores (e.g., \(f_i = f_j\)) while they are similar to each other, i.e., their similarity \(w_{ij}^{fp}\) is high. Hence, \(u_i\)’s normalized score will be high if follower \(u_j\) is an expert on \(Q_i\) (Intuition 1 (a)); meanwhile if \(u_i\) is similar to many followers specialized on \(Q_i\), she will be assigned a high ranking score (i.e., \(f_i\)), as \(f_i\) is proportional to \(D_u(i,i)\), which is the sum of similarities between \(u_i\) and her followers over the given topic \(Q_i\) (Intuition 1 (b)). Similar to the first term, the second term has the same purpose for lists (Intuition 3).

The third term (ref. Eq. (8)) is for the mutual ranking of users and lists. From the user perspective, if a user \(u_i\) is similar to her associated list \(L_j\), then user \(u_i\) and list \(L_j\) should be assigned similar normalized ranking scores, and the normalized score of \(u_i\) should be increased if \(L_j\) is relevant to \(Q_i\). If many lists containing \(u_i\) are relevant to \(Q_i\) and similar to \(u_i\) (i.e., \(D_{ml}(u_i)\) is large), \(u_i\)’s score (i.e., \(f_i\)) should be high (Intuition 2 (a-i)). From the list perspective, we can give similar analysis for lists.

By introducing the matrices \(S_u = (D_u)^{-\frac{1}{2}} W_u (D_u)^{-\frac{1}{2}}, S_l = (D_l)^{-\frac{1}{2}} W_l (D_l)^{-\frac{1}{2}},\) and \(S_{ml} = (D_{ml})^{-\frac{1}{2}} W_{ml} (D_{ml})^{-\frac{1}{2}},\) we can convert Eq. (8) into matrix-vector form as follows:

\[
\mathcal{F}(W_u, W_l, W_{ml}; f, g) = \alpha_1 f^T (I_n - S_u) f + \alpha_2 g^T (I_n - S_l) g + \alpha_3 (f^T f + g^T g - 2f^T (S_{ml}) g),
\]

where \(I_n\) is a \(n\)-by-\(n\) identity matrix. (9)

### 4.3.3 Loss Term

The regularization term does not incorporate the supervision information derived from the wisdom of Twitter crowds in the ranking process. Here, we proposed to use two types of relations as the supervised information for our problem, namely, MEM-OF relation and SUB-TO relation. The former is viewed as the supervision from the creators of lists. Intuitively a user listed in many relevant lists under a given topic is very likely to be an expert on that topic (Intuition 2 (a-i)). Similarly, a list that has many subscribers who are experts on a given topic is likely related to that topic (Intuition 2 (b)). Hence, we introduce two different indicator matrices to encode these two relations respectively.

Let \(n\)-by-\(m\) indicator matrix (i.e., \(X\)) encode the MEM-OF relations for supervising the ranking of users, and \(m\)-by-\(n\) indicator matrix (i.e., \(Y\)) encode the SUB-TO relations for supervising the ranking of lists. In particular, each element \((x_{ij})\) of \(X\) is set by \(x_{ij} = 1\) if \(u_i\) is the member of \(L_j\) (\(|u_i|\) is the number of lists containing \(u_i\)), and each element \((y_{ij})\) of \(Y\) is set by \(y_{ij} = 1\) if \(L_j\) is a list of \(u_i\) (\(|L_j|\) is the number of users who subscribe to \(L_j\)). Then, the loss term is defined as

\[
\ell = \begin{cases} 
(a) & \sum_{i,j} x_{ij} f_i g_j, & f_i \in \mathbb{R}, \quad x_{ij} \in \{0, 1\}, \\
(b) & \sum_{i,j} y_{ij} f_i g_j, & f_i, g_j \in \mathbb{R}, \quad y_{ij} \in \{0, 1\}, \quad f_i \in \mathbb{R}.
\end{cases}
\]

Here loss term (a) aims to ensure a user should be ranked higher if most of lists containing that user are related to the given topic. Similarly, loss term (b) aims to ensure a list should be assigned a higher ranking score if most of subscribers of that list are relevant to the given topic, which in return enhances the ranking scores of the associated members. Therefore, the loss can be formulated as follows,

\[
t(X, Y, f, g) = \gamma \left( \sum_{i=1}^{n} (f_i - \frac{1}{m} \sum_{j=1}^{m} x_{ij} g_j)^2 + (1 - \gamma) \sum_{j=1}^{m} (g_j - \frac{1}{n} \sum_{i=1}^{n} y_{ij} f_i)^2 \right),
\]

where \(\gamma (0 \leq \gamma \leq 1)\) is a non-negative coefficient (11) to trade-off the two different loss terms. Correspondingly, Eq. (11) can also be transformed into a matrix-vector form as follows

\[
t(X, Y, f, g) = \gamma \| f - Xg \|_2^2 + (1 - \gamma) \| g - Yf \|_2^2,
\]

where \(\| \cdot \|_2\) denotes \(l_2\)-norm of vector \(v\).

For ease of explanation, we use \(J(f, g)\) to denote the objective function in Eq. (7). By substituting Eq. (9) and (12), the optimization problem of this paper is formulated as follows:

\[
\min_{f \geq 0, g \geq 0} \left( \mathcal{F}(W_u, W_l, W_{ml}; f, g) + (1 - \lambda)\ell(f, g) \right)
\]

\[
\ell = \gamma \left( \sum_{i=1}^{n} (f_i - \frac{1}{m} \sum_{j=1}^{m} x_{ij} g_j)^2 + (1 - \gamma) \sum_{j=1}^{m} (g_j - \frac{1}{n} \sum_{i=1}^{n} y_{ij} f_i)^2 \right)
\]

\[
\min_{f \geq 0, g \geq 0} \left( \mathcal{F}(W_u, W_l, W_{ml}; f, g) + (1 - \lambda)\ell(f, g) \right)
\]

\[
\ell = \gamma \| f - Xg \|_2^2 + (1 - \gamma) \| g - Yf \|_2^2
\]

1041-4347 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Algorithm 1: Learning Global Authority Algorithm

1. **Input**: \( T : \{ t_i \} \): Twitter corpus, \( t_i \) denotes a appeared term in Twitter; \( \ell \): Candidate user set; \( E_u \): Follower graph, each entry in \( E_u \) denotes two users that follow each other; \( E_m \): Member graph, each entry in \( E_m \) denotes a member and her included list; \( E_{sc} \): Subscriber graph, each entry in \( E_{sc} \) denotes a subscriber and her subscribing list; \( E_{Li} \): List Graph, each entry in \( E_{Li} \) denotes two lists that are similar to each other; \( \alpha, \gamma, \rho, \varepsilon \)

2. **Result**: Authority matrix \( R \)

3. **Initialize**: Compute \( f^{(0)} \leftarrow \frac{-N(t_i, d^2)}{\sum_{t_j \in T} N(t_j, d^2)} \) and \( g^{(0)} \leftarrow \frac{-N(t_i, d^2)}{\sum_{t_j \in C} N(t_i, d^2)} \)

4. **Construct topic-specific matrices** \( W_{ui}, W_{wi} \) and \( W_{ml} \) by Eq. (6) and indicator matrices \( X \) and \( Y \)

5. **Set** \( s = 0 \)

6. **while** true do

7. **Compute** \( \nabla f^{(s)} = \frac{\partial f^{(s)}}{\partial \Phi} \) and \( \nabla g^{(s)} = \frac{\partial g^{(s)}}{\partial \Phi} \) by Eq. (14), (15)

8. **Update**: \( f^{(s+1)} \) and \( g^{(s+1)} \) by Eq. (16), (17)

9. **Normalize**: \( f^{(s+1)} \leftarrow \frac{f^{(s+1)}}{\sum_{j=1}^{n} f^{(s+1)}_{(j)}} \) and \( g^{(s+1)} \leftarrow \frac{g^{(s+1)}}{\sum_{j=1}^{n} g^{(s+1)}_{(j)}} \)

10. **Calculate**: \( J(f^{(s+1)}, g^{(s+1)}) \) by Eq. (13)

11. **if** \( J(f^{(s+1)}, g^{(s+1)}) - J(f^{(s)}, g^{(s)}) \leq \varepsilon \) **then**

12. **Return** \( R \)

13. **Stop**

Theorem 1 guarantees that objective function \( J(\hat{f}, \hat{g}) \) has first-order partial derivatives, which means the objective function at least has an optimal solution. Thus we introduce the gradient descent method to minimize \( J(\hat{f}, \hat{g}) \). The partial derivatives of \( J(\hat{f}, \hat{g}) \) with respect to \( \hat{f} \) and \( \hat{g} \) can be calculated as follows,

\[
\nabla \hat{f} = \frac{\partial \hat{f}}{\partial \Phi} = ((1 - \alpha + C_1) \Phi_X - \alpha \Phi_S + C_2 Y^T Y) \hat{f},
\]

\[
\nabla \hat{g} = \frac{\partial \hat{g}}{\partial \Phi} = ((1 - \alpha + C_2) \Phi_m - \alpha \Phi_{Sm} + C_1 X^T X) \hat{g} - (\alpha \Phi_{Sm} + C_1 X^T X + C_2 Y)^T \hat{g},
\]

where \( \rho \) is the step size that is allowed to change at each iteration, and \( s \) is the iteration number. When \( J \) is a convex function, all local minima can also be treated as the global minima. Consequently, our method can converge to a global solution. The processing of global authority of users on a given topic is summarized in Algorithm 1.

**Time Complexity**. The learning of global authority of users on a given topic is calculated offline. In fact, the computation overhead of SSGR is not high due to the sparsity of the matrices \( X, Y, S_u, S_i, \) and \( S_{ml} \). It can be solved with the iterative update rules (i.e., Eq. (14) and (15)) until convergence. We note that the entries of \( X^T X \) and \( Y^T Y \) remain unchanged for any topic, which is solely related to the graph structure, and thus they can be pre-computed. Therefore, the time complexity is only \( O(Km^2) \), where \( n \) is the number of users, \( m \) is the number of lists, \( K \) is the number of
iterations (which usually converges in fewer than 30 on our experimental datasets).

5 ONLINE RANKING FOR EXPERT FINDING

We propose a Gaussian-based method to estimate the local relevance of candidates on a given topic in Section 5.1. Then, in Section 5.2 we give an online ranking model, called RM, to address the expert finding problem by leveraging the global authority and local relevance of candidates on any topic.

5.1 Local Relevance Estimation

According to Eq. (4), for each pair of adjacent terms, \( t_j \) and \( t_{j+1} \), in query \( Q \), we estimate the local relevance of a user \( u_k \) to them as follows,

\[
K(t_j, t_{j+1}; d^u_k) = \Pr(t_j | d^u_k) \Pr(t_{j+1} | d^u_k) \exp \left(-\frac{1}{2} \frac{\Pr(t_j | d^u_k) - \Pr(t_{j+1} | d^u_k)^2}{N(t, d)}\right)
\]

where \( \Pr(t | d) = \frac{N(t, d)}{\sum_{d'} N(t, d')} \), and \( N(t, d) \) denotes the number of occurrences of term \( t \) in document \( d \).

The first two terms in the right hand of Eq (18) are used to favor users who frequently use the terms of query \( Q \). However, they cannot model the co-occurrences of \( t_j \) and \( t_{j+1} \). Hence, a Gaussian-based function (cf. the last term of Eq. (18)) is used to favor users who might frequently use two consecutive terms of query \( Q \).

5.2 Online Ranking Model

Next, we will present an online ranking model to address the Twitter expert finding problem for the arbitrary topic query (i.e., \( Q = \{t_1, \ldots, t_{|Q|}\} \)), which is defined based on Eq. (4) as follows,

\[
U_Q \leftarrow \arg \max_{u_k \in \mathcal{U}} \sum_{t_i \in t} K(u_k, t_i) \quad \text{if} \quad |Q| = 1;
\]

\[
\sum_{t_i \in Q} \left[ \sum_{i=1}^{|Q|} \Pr(t_i | d^u_k) \Pr(t_i | d^u_k) \right] K(t_i, t_{i+1}; d^u_k), \text{otherwise},
\]

where \( U_Q \) denotes the retrieved top-N experts that are most relevant to query \( Q \); \( \mathcal{U}(u_k, t_i) = R_{i,k} \), which is an entry of the authority matrix \( \mathcal{R} \). It indicates the global authority of user \( u_k \) on \( t_i \) computed by SSGR; function \( K \) is computed by Eq. (18). The online expert finding algorithm is given in Algorithm 2.

Time Complexity. Note that each row \( \mathcal{R}_i \) in \( \mathcal{R} \) is computed offline and the probability of term \( t_i \) in the context \( d^u_k \) of each user (i.e., \( \Pr(t_i | d^u_k) \)) can also be pre-computed offline. The time complexity of RM is \( O(n|Q|) \), where \( n \) is the number of users and \( |Q| \) is the length of the given topic query. In our experiments conducted on a real-world Twitter user set (about 0.5M users), the average running time of finding experts on a given topic is less than 0.01 seconds. The experiments are completed on a modest commodity desktop that is equipped with an Intel-i5 Dual-core 2.8GHz CPU and 8GB RAM. It shows that our proposed online expert finding model is computationally feasible for the real-time online Twitter expert finding applications.

6 EXPERIMENTS

6.1 Data Set

Users. The data set used in this paper was crawled via Twitter API\(^4\) from April 4, 2013 to June 10, 2013. For each user in Twitter, we crawled five types of data, i.e., user profiles, followers, tweets\(^5\), user-list membership information, and user-list subscribe information. In particular, we used a user-centric strategy to collect data as a brute-force crawling of all users for all lists would be prohibitively expensive and would not scale\(^6\) [5]. More specifically, to be unbiased to the users, we randomly crawled the information of users by utilizing a publicly available user collection\(^7\) as the seed set. Consequently, we obtained about 5.5M lists and 770,235 users who had subscribed to (or been members in) at least one list. In the 5.5M lists, 73.44% lists only had a List name while the others had a description (detailed statistics is in Table 2). In addition, the dataset contained a mixture of different languages, e.g., Chinese, English, German, Italian and etc. We filtered non-English characters, stopwords, punctuation as well as the high-frequency words in Twitter (e.g., “RT”), and employed Porter’s stemmer [35] for remaining words. After the processing, the users without any context information are removed; finally we obtained 491,622 users (with 61.6M tweets and 4.4M lists) out of 770,235 users as the experimental dataset, named TwL (i.e., Twitter-List). The details about TwL are shown in Table 3.

Queries. We use 28 sample queries for evaluation, whose topics are from general to specific, e.g., a general personal hobby like “traveling” or a specific Top-News like “Boston Marathon bombings”, which can be used to comprehensively evaluate the effectiveness

5. For each user, we collect a set of the most recent (\( \leq 1000 \)) posted tweets.
6. Twitter normally rate-limited the number of API requests from a single machine (IP Address) to 150 per hour, i.e., 3600 user profile crawls per day.
TABLE 2: Statistics of the user-list relations in the crawled data.

<table>
<thead>
<tr>
<th># Lists</th>
<th># Lists (No Description)</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEM-OF (A1)</td>
<td>5,410,831</td>
<td>3,090,655</td>
</tr>
<tr>
<td>SUB-TO (A2)</td>
<td>635,852</td>
<td>340,616</td>
</tr>
<tr>
<td>A1 U A2</td>
<td>5,563,355</td>
<td>3,437,271</td>
</tr>
</tbody>
</table>

TABLE 3: Statistics of TwLI. Each user in TwLI has subscribed to (or been members in) at least one list.

<table>
<thead>
<tr>
<th>User</th>
<th>Total #</th>
<th># M</th>
<th># S</th>
<th># M ∩ S</th>
</tr>
</thead>
<tbody>
<tr>
<td>491,622</td>
<td>452,119</td>
<td>121,449</td>
<td>89,946</td>
<td></td>
</tr>
<tr>
<td>4,486,954</td>
<td>4,412,514</td>
<td>401,780</td>
<td>387,340</td>
<td></td>
</tr>
</tbody>
</table>

TABLE 4: Sample queries used for evaluation.

<table>
<thead>
<tr>
<th>Category</th>
<th>Sample Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>News</td>
<td>Egypt Balloon Explosion, Iran Nuclear Program, Curiosity on Mars, Boston Marathon bombings, Fukushima nuclear leak.</td>
</tr>
<tr>
<td>Sports</td>
<td>football, soccer</td>
</tr>
<tr>
<td>Hobbies</td>
<td>traveling, photography, cooking, classical music</td>
</tr>
<tr>
<td>Science</td>
<td>biology, computer science</td>
</tr>
<tr>
<td>Entertainment</td>
<td>classical music</td>
</tr>
<tr>
<td>Lifestyle</td>
<td>dining, wine, health, fashion</td>
</tr>
<tr>
<td>Technology</td>
<td>smartphone, data mining, apple app, linux, cloud computing, iphone</td>
</tr>
<tr>
<td>Business</td>
<td>stock, finance, markets, energy</td>
</tr>
</tbody>
</table>

6.2 Experimental Setting

6.2.1 Ground Truth & Evaluation Metrics

Ground Truth. To evaluate the quality of the expert search results of different methods, we follow the evaluation strategy in [5], [4]. That is, we aggregate the top-10 users returned by each evaluated method, and then 9 graduate students (whose research areas are not in text processing area) are invited for labeling. The annotators are divided into 3 groups (3 annotators in each group) to label each suggested user (as shown in Fig. 4). Each user is labeled to be relevant (score 1) or irrelevant (score 0) with respect to the given query by evaluators. In each group if conflicts happen, the third annotator determines the final result of each group, and the majority vote of groups is used as the label of the user. Each evaluator is required to label the relevance of users based on the contents of their posted tweets (e.g., whether including the URLs related to the given query), users’ bios and the metadata of lists containing that user.

Evaluation Metrics. To evaluate the expert finding performance of different approaches, we adopt the following evaluation metrics: (i) Precision [29] (P@N). It measures the percentage of relevant user in the top-N returned users. (ii) Normalized Discounted Cumulative Gain [29] (NDCG@N). It measures the performance of expert finding system based on the relevance (i.e., relevant (1)/irrelevant (0)) of the selected experts, which is the normalization of Discounted Cumulative Gain (DCG) at each position for a chosen value of k. In our experiments, we use P@5, P@10, NDCG@5 and NDCG@10.

8. The annotated results can be accessed from the following link: https://www.dropbox.com/s/47/up13gqcrb7zz/annotated_new.txt?dl=0.


Fig. 4: Non-anonymous label screen shows bio, tweets (≤ 3) and lists (≤ 4) of a user and asks evaluators to label for relevant or irrelevant to a given topic query.

6.2.2 Baseline Methods & Parameter Setting

We compare our approach with TwitterRank [3] and Cognos [5]. In [5], Cognos is demonstrated to outperform the other previous state-of-the-art methods, such as [4] that relies on the user’s bio or tweets, and WTW [3] (Twitter Who To Follow) that is the official Twitter expert search service. We evaluated 7 expert finding methods listed in Table 5.

TwitterRank [3]. This method first employs Lateral Dirichlet Allocation (LDA) [30] model to identify users’ interested topics from their tweets, then builds a topic-specific graph for each detected topic to compute a PageRank [34] vector of users, and finally linearly combines the PageRank vectors of different topics of a given query for finding topic-specific influential users. The damping factor of PageRank algorithm is set at 0.85 by following [3].

Cognos [5]. This method employs a topic vector to represent each Twitter user by the Twitter List information of users (MEM-OF relation) and uses cover density ranking (CDR) [31] method to identify topic-specific experts. The length of covers (|X|) is selected from {4, 6, 8, 10, 12, 14, 16} and |X| with the best performance is used to report the final comparison results.

Our Method. Our proposed approach chooses the topic-specific authorities by jointly exploiting users’ profiles and the meta-data of lists containing users, as well as the three different types of relations, i.e., follower relation, list-list relation and user-list relation. Our approach contains a graph-based ranking method
TABLE 5: Different comparison methods.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PT</strong></td>
<td>User’s profiles (bios and tweets).</td>
</tr>
<tr>
<td><strong>PL</strong></td>
<td>User’s bio and the meta-data of lists containing users.</td>
</tr>
<tr>
<td><strong>PTL</strong></td>
<td>User’s profiles (bios and tweets) and the meta-data of lists containing users.</td>
</tr>
<tr>
<td><strong>Methods For Comparison</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Cognos</strong></td>
<td>PageRank-based method [3] (PT)</td>
</tr>
<tr>
<td><strong>RM-PL</strong></td>
<td>Online Ranking model (RM) based on $R_{i,k}$ (PL)</td>
</tr>
<tr>
<td><strong>RM-PTL</strong></td>
<td>Online Ranking model (RM) based on $R_{i,k}$ (PTL)</td>
</tr>
<tr>
<td><strong>SSGR-RM-PTL</strong></td>
<td>Online Ranking model (RM) based on SSGR (PTL)</td>
</tr>
<tr>
<td><strong>SSGR-RM-PL</strong></td>
<td>Online Ranking model (RM) based on SSGR (PL)</td>
</tr>
</tbody>
</table>

(i.e., SSGR) and a ranking model (i.e., RM). Correspondingly, our method is denoted by SSGR+RM.

To analyze the different impacts of the two parts of our approach, the online ranking model (RM) is formulated as a baseline method by replacing the global authority score ($R_{i,k}$) computed by SSGR with $R_{i,k}$, which is calculated by a straightforward method. Here, $R_{i,k}$ is the similarity between the $i$-th term $t_i$ and the $k$-th user in the corpus, and is computed by $R_{i,k} = \sum_{j \in c(t_i)} N(t,d_j)$, where $N(t,d)$ denotes the number of occurrences of term $t$ in document $d$. In addition, we have three different methods of constructing the information of users: (i) **PT**, user’s bio and tweets; (ii) **PL**, user’s bio and the meta-data of lists containing those users; and (iii) **PTL**, user’s bio and tweets, as well as the meta-data of lists containing users. Correspondingly, different variants of SSGR+RM and RM are listed in Table 5.

In addition, there are some other works on the detection of influential users in social networks, such as Community Question Answering [12], [11], [42], Blog [27], Academic Social Network [23], Twitter [33], [43], and other social networks [38], [41]. However, these methods utilize different domain features (e.g., the category information in CQA domain) or do not consider topical dimension, which are thus not appropriate to make a comparison of them with the current work.

The parameters for our proposed method are empirically set as follows: $\alpha_1 = 0.33$, $\alpha_2 = 0.33$, $\alpha_3 = 1 - \alpha_1 - \alpha_2$, $\theta = 0.1$, $\delta = 10^{-8}$, $\rho = 0.1$, $\varepsilon = 10^{-12}$, and $\lambda = \gamma = 0.5$. As the construction of list graph is based on mutual k-nearest neighbor graph [39], we empirically set $k = 50$ in this paper.

6.3 Evaluation Results and Analysis

Comparison of the Expert Finding Performance:

This experiment is to evaluate the effectiveness of finding the topic-specific experts by our approach, i.e., SSGR+RM-PTL. In this work, we compare SSGR+RM-PTL with the baseline methods TwitterRank and Cognos. Fig. 5(a)-(b) shows P@5, P@10, NDCG@5 and NDCG@10 of each method.

From Fig. 5(a)-(b), we observe that: First, Cognos performs better than TwitterRank on all metrics. For example, Cognos outperforms TwitterRank by 30.19% ($p$-value $\leq 0.005$) in terms of Precision@5. This is because Cognos utilizes Twitter List relation to find the topic-specific authorities while Twitterrank is based on the propagation (reciprocity in follower relations [3]) of the topical importance of users in follower graph. The results demonstrate that methods using Twitter Lists (Cognos) is more effective than approaches to utilizing follower graph and user’s tweets (TwitterRank). Second, our proposed method SSGR+RM-PTL consistently outperforms the two baseline methods. The improvements are statistically significant on all metrics ($p$-value $\leq 0.005$). For example, SSGR+RM-PTL outperforms TwitterRank by 75.68% ($p$-value $\leq 0.000001$) and 19.23% ($p$-value $\leq 0.005$), Cognos by 22.64% ($p$-value $\leq 0.01$) and 16.58% ($p$-value $\leq 0.005$), in terms of Precision@5 and NDCG@5, respectively. The reason might be due to two facts: (i) Unlike TwitterRank that employs PageRank algorithm [34] on follower relation graph or Cognos that employs a similarity measure [31] to rank users based on user-list relation, SSGR+RM-PTL effectively exploits three different types of relations among users and lists. Specifically, it employs a normalized Laplacian regularization to take into account different relations (i.e., follower relation, user-list relation and list-list relation) for ranking, and utilizes user-list relations reflecting the wisdom of Twitter crowds to supervise ranking users. (ii) SSGR+RM-PTL makes use of two types of user-related information to model user’s domain of expertise, i.e., user’s profiles and List information. On one hand, some queries almost do not appear in Lists, such as the top news “Boston Marathon bombings”[^6], which is more likely contained in the user’s tweets. On the other hand, Lists are usually carefully built according to the wisdom of Twitter crowds, which are trustworthy for identifying the topical expertise of users contained in the lists. Our method SSGR+RM-PTL is able to make use of both types of information. The results demonstrate the effectiveness and superiority of our proposed method as compared to the state-of-the-art method Cognos and TwitterRank.

Our proposed approach consists of two parts, i.e., a graph-based ranking method (SSGR) and a ranking example.

The Impact of Graph-based Ranking Method. This is to evaluate the effectiveness of the graph-based ranking method in our approach. Since the graph-based ranking method cannot work alone for expert finding, we compare the methods utilizing the ranking scores computed by SSGR (i.e., SSGR+RM-PT, SSGR+RM-PL and SSGR+RM-PTL) and methods without using such ranking scores (i.e., RM-PL and RM-PTL), as well as TwitterRank. The Precision@N and NDCG@N of each method are plotted in Fig. 6(a)-(b).

As can be observed from Fig. 6(a)-(b): (i) Methods utilizing the ranking scores of SSGR outperform the methods without using such ranking scores. In terms of Precision@5, SSGR+RM-PL and SSGR+RM-PTL outperform RM-PL and RM-PTL by 11.93% (p-value ≤ 0.05) and 8.33% (p-value ≤ 0.05), respectively. (ii) Even without using the meta-data of Twitter Lists to model user’s information, our proposed method (i.e., SSGR+RM-PT) still outperforms TwitterRank. The improvements are statistically significant on all metrics (p-value ≤ 0.05). For example, SSGR+RM-PT outperforms TwitterRank by 31.21% (p-value ≤ 0.005) and 11.93% (p-value ≤ 0.05) in terms of Precision@10 and NDCG@10, respectively.

The observation (i) and (ii) demonstrate the effectiveness of our proposed graph-based ranking method SSGR, in exploiting different relations among users and lists for identifying the users’ domain of expertise.

The Impact of Online Ranking Model. This experiment is to evaluate the effectiveness of the ranking model (RM). Next we use two groups of experiments to further evaluate the impact of each part in our approach. The first group is to evaluate the impact of the graph-based ranking method SSGR; and the second group is to evaluate the impact of the ranking model RM.

As shown in Fig. 7(a)-(b), even utilizing the same strategy for constructing the information of users (i.e., PL), RM-PL performs better than Cognos in terms of Precision@5, NDCG@5 and NDCG@10 (except Precision@10). The reason might be RM considers not only the local relevance between a user and the given query, but also the global authority score (i.e., $R_{ij}^*$) of that user on the given query. Additionally, RM is also capable of effectively finding experts by: (i) combining with other global authority scores. For example, by replacing $R_{ij}^*$ with the ranking score of SSGR, RM (i.e., SSGR+RM-PL) consistently outperforms Cognos on all metrics; and (ii) utilizing other information. For instance, with PTL, RM (i.e., RM-PTL) outperforms Cognos by 13.56% (statistically significant, p-value ≤ 0.01) in terms of NDCG@5.

6.4 On the Sensitivity of Parameter

In this subsection, we study the impact of parameters in our method.

First, we study the impact of parameter $\lambda$, which is used in Eq. (13) to trade-off the regularization term and the loss term. We compare 3 different variants of our proposed methods when varying $\lambda$ from 0 to 1, i.e., SSGR+RM-PT, SSGR+RM-PL and SSGR+RM-PTL. As shown in Fig. 8(a)-(b), the performance of our methods do not significantly change with varying $\lambda$. We observe that our proposed methods achieve the best result when $\lambda$ is within the range of $[0.3 - 0.5]$, and simultaneously making use of both regularization term and loss term outperforms the extreme cases when only the regularization term ($\lambda = 1$) or the loss term ($\lambda = 0$) is used.
Second, we study the importance of different types of relations (i.e., user-user relation, user-list relation and list-list relation) in our approach. In particular, as our method is to mutually reinforce the ranking of users and lists by means of the three relations, to learn the global authority of users on a given topic. Hence, we fix the parameter $\lambda$ at 1, and vary each of the three parameters $\alpha_1$, $\alpha_2$ and $\alpha_3$ to evaluate the impact of each type of relations in our method (i.e., SSGR+RM-PTL). In each time, the other two parameters are set at $\frac{1-\alpha}{2}$. For example, the value of parameter $\alpha_1$ and $\alpha_3$ are set at $\frac{1-\alpha}{2}$ when varying $\alpha_2$. Fig. 9(a)-(b) shows Precision@10 and NDCG@10 of our method (i.e., SSGR+RM-PTL) with respect to different $\alpha$ (i.e., $\alpha_1$, $\alpha_2$ and $\alpha_3$) ranging from 0.1 to 0.9 with an increment of 0.1. As shown in Fig. 9, when varying one of the three parameters (i.e., $\alpha_1$-$\alpha_3$) respectively, the performance first increases and then decreases. The average improvements of varying $\alpha_3$ over varying $\alpha_1$ and varying $\alpha_2$ are (27.89%, 2.98%) and (38.63%, 5.28%), in terms of Precision@10 and NDCG@10. This demonstrates the importance of user-list relation, which contributes more to improve the performance of our method, as the correlation between user-user relation and list-list relation are bridged by user-list relation. From Fig. 9(a)-(b), we can observe that our proposed method achieves the best performance with setting $\alpha_3$ at 0.6, and the best parameter setting for our method is: $\alpha_1 = 0.3$, $\alpha_2 = 0.1$ and $\alpha_3 = 0.6$.

6.5 In-depth Analysis of Expert Search Results

In this subsection, we present an in-depth analysis of expert search results of our proposed approach.

Single Term Query. We observe that both SSGR+RM-PTL and Cognos achieve comparable performance over most of single term queries while TwitterRank performs slightly worse. For example, for query “astronomy”, both SSGR+RM-PTL and Cognos find the same user “NASA” and TwitterRank selects “hubble-science” on the top of their ranking lists. However, for 7 queries on hot topics (e.g., “stock”, “finance”, etc.), TwitterRank and SSGR+RM-PTL perform worse than Cognos. This is because single term queries on hot topics are frequently contained in many users’ tweets, and our method utilizing user’s tweets and user’s follower relations, i.e., SSGR+RM-PTL, would choose these users if they post many relevant tweets and have many relevant followers; however these users may not be experts on such topics. For example, for query “stock”, SSGR+RM-PTL selects user ‘Bertieis’ as a top-10 result, who publishes many tweets containing “stock”, and has 229 followers (most of them publish tweets containing “stock”) in our experimental data set TwL. Interestingly, she is also included in a list named “Stocks”. However, she is not an expert on stock. In contrast, Cognos is less affected as it chooses users contained by many relevant lists.

Multiple Term Query. We observe that SSGR+RM-PTL consistently outperforms Cognos and TwitterRank for all multiple term queries$^{11}$, which are usually more specific than single term queries. For example, for query “classical music”, Cognos selects many pop singers (e.g., “AvrilLavigne”) and TwitterRank chooses users with many music-related followers (e.g., “thebowradio”) while SSGR+RM-PTL selects users like “nyphil”, which is the official Twitter account of New York Philharmonic.

To better illustrate the effectiveness of our proposed method on the multiple term queries, the top-6 (due to the space limitation) selected users returned by SSGR+RM-PTL, Cognos and TwitterRank for two representative queries are shown in Table 6 and Table 7, respectively. From the results, we observe that: (i) SSGR+RM-PTL is able to effectively choose the topic-specific experts for queries with specific topics. For example, in addition to “mwalhby”, all other users found by our method in Table 6 have posted the tweets regarding the topic “Egypt Balloon Explosion$^{12}$”. In contrast, none of users found by Cognos or TwitterRank has reported that topic. (ii) Unlike Cognos, which tends to select users listed by many other users, our approach fairly treats all Twitter users for search. For example, user “JoAnnaScience” in Table 7, who has only 4 Lists, is also found by SSGR+RM-PTL, and it is however not selected by Cognos. (iii) Our proposed method is more robust than Cognos and TwitterRank. For example, in Table 7, Cognos chooses some irrelevant users (e.g., “SmiTomEchoelon”) for the topic “Curiosity on Mars”. This is because there are many Twitter lists are built by the fans of an American rock band named “Thirty Seconds to Mars$^{13}$”. However, all of users chosen by SSGR+RM-PTL are relevant to the given query. Similarly, in Table 6, TwitterRank selects several users relevant to “Balloon” for query “Egypt Balloon Explosion”, as TwitterRank employs a linear function to combine the PageRank scores of users on different topics in the given query. In contrast, most of users (exclude “mwalhby”) found by SSGR+RM-PTL are relevant to the query.

7 Conclusions

In this paper, we address the problem of topic-specific expert finding in Twitter. We successfully integrate different types of user-related information (i.e., the crowdsourced Lists information, follower graph and users’ profiles) into a unified ranking framework for accurately inferring the topical expertise of users. To the best of our knowledge, this is the first attempt that targets expert finding problem in Twitter by making use of all of such information. Specifically, within

\[11.\] 39% queries are multiple term query in Table 4.


TABLE 6: Top-6 experts selected for query “Egypt Balloon Explosion”, along with users’ extracted bios and tweets in TwL. Relevant experts are highlighted in bold font.

<table>
<thead>
<tr>
<th>User</th>
<th>Extracted Bio (B) and related tweets (T)</th>
<th>List</th>
<th># List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bennu</td>
<td>[B] Deliverying daily breaking news about ...</td>
<td>Egyptian</td>
<td>371</td>
</tr>
<tr>
<td>Scorpion-kiss</td>
<td>[B] Documentary producer...</td>
<td>Egyptian</td>
<td>23</td>
</tr>
<tr>
<td>Alternative-Egypt</td>
<td>[T] Hot air balloon crash in Egypt kills 19 tourists...</td>
<td>Egyptian</td>
<td>34</td>
</tr>
<tr>
<td>Breaking-NZ</td>
<td>[B] Biggest alerts... Updates delayed by 10 minutes...</td>
<td>Egyptian</td>
<td>57</td>
</tr>
<tr>
<td>Egyptian-Terror</td>
<td>[T] The latest Egyptian news and discoveries...</td>
<td>Egyptian</td>
<td>113</td>
</tr>
<tr>
<td>mwabby</td>
<td>[B] Microsoft Egypt Program Manager...</td>
<td>Egyptian</td>
<td>4</td>
</tr>
</tbody>
</table>

Cognos Results

<table>
<thead>
<tr>
<th>User</th>
<th>Extracted Bio (B) and related tweets (T)</th>
<th>List</th>
<th># List</th>
</tr>
</thead>
<tbody>
<tr>
<td>shahidhamid</td>
<td>Director of Research... Middle East Policy...</td>
<td>Egypt</td>
<td>1,838</td>
</tr>
<tr>
<td>dufaormale</td>
<td>Egypt’s largest news organization...</td>
<td>Egypt</td>
<td>1,231</td>
</tr>
<tr>
<td>Linnaiah</td>
<td>Journalist...</td>
<td>Egypt</td>
<td>498</td>
</tr>
<tr>
<td>NevineZaki</td>
<td>Writer &amp; host of ‘Movie Pictures’...</td>
<td>Egypt</td>
<td>376</td>
</tr>
<tr>
<td>Elaluz</td>
<td>I do not claim to be objective, subjective...</td>
<td>Egypt</td>
<td>244</td>
</tr>
<tr>
<td>stevenacook</td>
<td>Senior fellow for Middle Eastern studies...</td>
<td>Egypt</td>
<td>464</td>
</tr>
<tr>
<td>CarbinCropy</td>
<td>The Balloon Bandit of Amusement...</td>
<td>Egypt</td>
<td>49</td>
</tr>
<tr>
<td>BalloonFDN</td>
<td>Build membership for the ABO Balloon Museum...</td>
<td>Egypt</td>
<td>49</td>
</tr>
<tr>
<td>tutti</td>
<td>Traveler...</td>
<td>Egypt</td>
<td>133</td>
</tr>
<tr>
<td>ncserspace</td>
<td>NC Near Space Research...</td>
<td>Egypt</td>
<td>10</td>
</tr>
<tr>
<td>VOAArrest</td>
<td>Cairo Bureau chief and regional correspondent...</td>
<td>Egypt</td>
<td>27</td>
</tr>
<tr>
<td>musicnever</td>
<td>Norwegian musician artist Helge Krabbe (Homeless Balloon)...</td>
<td>Egypt</td>
<td>6</td>
</tr>
</tbody>
</table>

TABLE 7: Top-6 experts selected for query “Curiosity on Mars”, along with users’ extracted bios and tweets in TwL. Relevant experts are highlighted in bold font.

<table>
<thead>
<tr>
<th>User</th>
<th>Extracted Bio (B) and related tweets (T)</th>
<th>List</th>
<th># List</th>
</tr>
</thead>
<tbody>
<tr>
<td>martian-soil</td>
<td>[B] All the fresh dirt on the planet Mars...</td>
<td>Mars</td>
<td>199</td>
</tr>
<tr>
<td>NASA</td>
<td>[T] Landing on Mars...</td>
<td>Mars</td>
<td>57,001</td>
</tr>
<tr>
<td>JoAnna-Science</td>
<td>[T] Science need and future science writer...</td>
<td>Mars</td>
<td>4</td>
</tr>
<tr>
<td>Mars-SanDiego</td>
<td>Curiosity arm...</td>
<td>Mars</td>
<td>34</td>
</tr>
<tr>
<td>mars-today</td>
<td>NASA Curiosity Rover Wins Prestigious Award...</td>
<td>NASA</td>
<td>303</td>
</tr>
<tr>
<td>TheSpace-Trap</td>
<td>Collecting amazing photos... on our universe...</td>
<td>NASA</td>
<td>187</td>
</tr>
</tbody>
</table>

Cognos Results

REFERENCES
