PIN: Potential Wise Crowd From Million Grassroots

Yao Wu, Tao Huang, Dan Zhao, Hong Chen*
Cuiping Li
Key Laboratory of Data Engineering and Knowledge Engineering of Ministry of Education, Beijing, China
School of Information, Renmin University of China, Beijing, China
{ideamaxwu,taohuang,danzhao,chong,licuiping}@ruc.edu.cn

ABSTRACT

Crowdsourcing proves a viable approach to solve certain large-scale problems by posting tasks distributively to humans and harnessing their knowledge to get results effectively and efficiently. Unfortunately, crowdsourcing suffers from lack of available participants with domain knowledge or skills. In this paper, we propose potential wise crowd (i.e., a crowd with similarity and diversity in domain knowledge) find from million grassroots in social networks. We design and implement a distant-supervision framework to find potential crowdsourcers from existing social networks. A knowledge graph is used to assess the domain knowledge in terms of similarity and diversity. The wise crowd formation is a NP-hard problem and we propose greedy algorithms to approach it. Experimental results show the performance of our framework and algorithms in aspects of effectiveness and efficiency.

CCS CONCEPTS

•Information systems → Specialized information retrieval; Specialized information retrieval; •Human-centered computing → Ubiquitous and mobile computing systems and tools; Ubiquitous and mobile computing systems and tools; Ubiquitous and mobile computing systems and tools; •General and reference → *Design;*

KEYWORDS

mobile crowdsourcing, distant supervision, crowd formation, mobile recruitment framework

1 INTRODUCTION

Mobile devices unfold the full potential of crowdsourcing, allowing users to transparently contribute to solving complex and novel problems. Users carrying with mobile devices can travel from places to places to collect various multimedia data, accomplish different tasks and provide qualified services. Prevail of crowdsourcing motivates us to better utilize the power of grassroots [1, 2].

Unfortunately, crowdsourcing suffers from several difficulties that prevent it from thriving. First, most crowdsourcing platforms lack enough participants due to a variety of reasons [3]. Without enough users, it is hard to guarantee the performance of crowdsourcing platforms. Second, many efforts rely on paid crowdsourcing, e.g., Amazon Mechanical Turk[1]. Users who are motivated by monetary rewards are often less self-devoted than the users who are unpaid and motivated by other means [4]. Third, workers participating in paid crowdsourcing are typically non-expert users, and often lack the domain knowledge needed for certain crowdsourcing efforts [5]. Therefore, it is necessary to find potential participants with domain knowledge in an effective and efficient way.

Involving crowds in performing tasks is an important aspect of many crowdsourcing systems and applications [6]. A lot of emphases have been given so far to address random and general grassroots for micro-task assignment on platforms such as Amazon Mechanical Turk. Selecting random workers on crowdsourcing platforms is an economic way for most expertise-independent problems, e.g., cat images labeling.

However, to certain tasks, a selected crowd with domain knowledge performs better. Take language translation for example, and only persons speaking or knowing the specific language can interpret comprehensively. Even in the image-labeling example, if the tasks are to tell particular species of birds rather than common cats, the domain knowledge is necessary. Except the similarity of domain knowledge, diversity plays an important role in the process of decision-making, and lack of diversity may result in potential opinion bias [7, 8].

Nowadays, the popularity of social networks provides an increasingly rich source of information about public opinion and current events, which can be valuable to professionals across a wide range of industries. Social networks data can be very useful for potential cowdsourcing participants discovery due to the variety of existing applications with meta data in content and diversity of users associated with. However, finding crowds from social networks with millions of users, such as Twitter, is not a trivial problem and the challenges lie in effectiveness and efficiency.

In this paper, we propose to find potential wise crowds from million grassroots in social networks. A wise crowd is a set of crowdsourcers with similarity and diversity in domain knowledge. Traditional approaches to identify topical experts rely either on the textual contents provided by the user him/herself (e.g., in the short (auto-) biography) or on analyzing the network characteristics and social activities of users. There are three drawbacks in current work: 1) they focus on a specific domain community, such as academic networks, which makes it uneasy to apply in other situations; 2) they cannot handle the mismatch problem well based on only keywords matching; 3) link based algorithms share a common problem:

Corresponding author.

MobiQuitous 2017, Melbourne, VIC, Australia

978-1-4503-5368-7/17/11…$15.00
DOI: 10.1145/3144457.3144458

[1] https://www.mturk.com/mturk/

topic drift, which makes the most in-links in the network tend to dominate.

From these observations, we propose to explore potential participants from existing social networks rather than designing incentive mechanisms [9] for recruitment of high qualified participants at a monetary cost. Given a query of task, we expect a set of crowdsourcers returned with similarity and diversity based on domain knowledge. In our work, we propose distant supervision based knowledge matching to assess the expertise and design greedy algorithms to form the wise crowd. To address the limitations of conventional techniques, the distant supervision approach exploits well-developed knowledge base (e.g., Google Knowledge Graph[2]) with wide coverage and high accuracy to facilitate the matching problem. Our contributions are summarized as follows.

- We design and implement a framework to find potential crowdsourcers from million grassroots in existing social networks scalably.
- We propose distant supervision based on knowledge graph to effectively assess the domain knowledge of potential crowdsourcers. We model the knowledge matching as a probabilistic representation.
- We find the problem of the wise crowd formation as NP-hard problem and propose greedy algorithms to approach it efficiently.
- Extensive experiments show the performance of our solution in aspects of scalability, effectiveness and efficiency.

The rest of the paper is organized as follows. We first review related works and situate our contributions in Section 2. Then we model the problems and present the framework in Section 3. The proposed knowledge assessment by distant supervision and wise crowd formation with greedy algorithms follow in Section 4 and Section 5, respectively. Afterwards, we evaluate the performance in Section 6. Finally, we conclude our work in Section 7.

2 RELATED WORK

Identifying users with domain knowledge can trace back to expert finding (expert search), which has been widely studied especially by the human-computer interaction (HCI) and information retrieval (IR) communities. First pioneering approaches to expert finding could be classified as *profile-centric* and the follow-up *document-centric* approaches analyze the content of each document separately. The latter generally performs better at ranking, while the former is more efficient as it avoids retrieving all documents relevant to a query. Lately, the *link-based* analysis emerges. Others like *activity based* are also under research.

Balog et al. [10] defined and compared two models: the candidate-based model and the document-based model. Tang et al. [11] study the problem of Web user profiling, which is aimed at finding, extracting, and fusing the user profile from the Web. Daud et al. [12] propose a novel Temporal-Expert-Topic (TET) approach based on Semantics and Temporal Information based Expert Search (STMS) for temporal expert finding, which simultaneously models conferences influence and time information. Deng et al. [13] investigate and develop two community-aware strategies to enhance expertise retrieval.

[2]https://www.google.com/intl/es419/insidesearch/features/search/knowledge.html

As social media becomes more prevalent, a better understanding of how expertise can be derived from social media data is vital and can contribute to the overall value-proposition of social media in the enterprise. Guy et al. provide an extensive study that explores the use of social media to infer expertise within a large global organization [14]. Quan et al. [15] present a novel model towards this goal by integrating topic modeling with short text aggregation during topic inference. Smirnova et al. [16] develop a Bayesian hierarchical model for expert finding that accounts for both social relationships and content. Zhang et al. [17] propose a propagation-based approach that takes into consideration both person local information and network information. Serdyukov et al. [18] suggested a graph-based approach for expert finding in large enterprises, which models the expert finding as a walking process in a graph of topical documents and related persons. Bozzon et al. [19] considers social networks both as a source of expertise information and as a route to reach expert users by considering their profiles and tracing their activities in social networks.

Ipeirotis et al. [20] propose to use existing Internet advertising platforms for targeting and attracting users. Cognos [21] takes an entirely different approach to identify topical experts in Twitter utilizing crowdsourced topical annotation of experts. Kang et al. [22] propose a framework to automatically identify experts based on the linguistic and structural features of the annotations they create, and use experts' annotations to guide the folksonomy learning process to reach diversity. Su et al. [23] investigate the task of diversifying expert finding in the context of academic social network and leverage supervised learning to learn a diversity retrieval function. Crowds are increasingly being adopted to solve complex problems while size and diversity are two key characteristics of crowds[24].

Prior works attempt to find experts, even in social media. But the try of utilizing distance supervision with knowledge graph to facilitate the crowd formation in crowdsourcing background is rare. Besides, most crowdsourcing focus on task assignment [25] based on strict constrains (e.g., budget) with an objective function. Domain knowledge with similarity and diversity is barely considered, neither to implement such a framework to find potential users from million grassroots.

3 FRAMEWORK OF WISE CROWD FIND

Motivating Example: We call for some participants who have knowledge of Big Data to do certain tasks. However, big data is a general term and the related sub-domains can be data ingestion, data storage and data analysis. Therefore, the returned users should be diverse enough to cover as many subtopics as possible without loss of similarity in domain knowledge. During the knowledge assessment, we extract the potential domain knowledge of users from their social contents and extend it by distant supervision. In the process of crowd formation, it is more challenging to find a wise group to cover the required knowledge types as well as to take similarity into consideration.

3.1 Fundamental Problem Formulation

Given a wise crowd query (a natural language query based on terms) and a group of social network users (take Twitter users

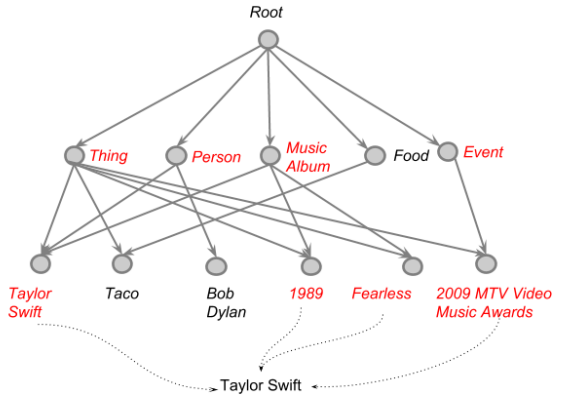


Figure 1: An example of the query "Taylor Swift" with corresponding linked entities and knowledge types from Google Knowledge Graph.

for example), the goal is to select a set of users from grassroots with domain knowledge to form the most "wise" crowd in terms of similarity and diversity to address the query. The problem can be divided into two subproblems: domain knowledge assessment and wise crowd formation. The knowledge assessment is to match the social network users $\mathcal{U} = \{u_1, u_2, ..., u_n\}$ to the knowledgeable turks [3] $\mathcal{T} = \{t_1, t_2, ..., t_n\}$ from their social media contents. The wise crowd formation is to form a set of targeted tukrs $\mathcal{T}^*$ from all extracted turks $\mathcal{T}$ to satisfy the requirements of similarity and diversity.

Knowledge Assessment. State-of-the-art expert retrieval methods usually estimate the relevance between the query and the documents of candidates using language model. However, the language model has a limitation that all the query terms should occur in each support document, by which some real experts cannot be searched. Typical topic modeling techniques such as LDA have demonstrated great successes on long documents, but unfortunately, they have not been able to work very well on short texts [26]. This is mainly due to the fact that only very limited word co-occurrence information is available in such short and sparse texts like tweets compared with long documents. Furthermore, not as simple as popular celebrities, the definition of experts introduces additional difficulties. The knowledge conveyed in what they post is essential. The challenges mentioned above inherently come from insufficient representations. They motivate us to propose a more flexible domain knowledge assessment solution to explore effective representations that are able to tackle the complexity that lies in the social media data.

To go beyond the feature-based classification methods and vector representation inference in expert finding, a potential solution is to incorporate the semantic information from knowledge graph. We achieve this goal by representing users via Google Knowledge Graph (we call it the Graph for brevity) to capture correlations among words and topics. Given a query q to the Graph, it returns the most relevant entities y_q with scores $s_{(y,q)}$ and the knowledge type z_q it attaches to.

[3]Turks refer to crowdsourcers named from mechanical turks when not confusable.

For example, given the query "Taylor Swift", the top five results from the Graph are {Taylor Swift (886.061462) @Thing @Person; Taylor Swift (524.517761) @MusicAlbum @Thing; 2009 MTV Video Music Awards (354.305084) @Event @Thing; Fearless (325.5325034) @MusicAlbum @Thing; 1989 (324.026611) @MusicAlbum @Thing} with corresponding entities and types, illustrated by Fig. 1. The Graph is much more complicated as Fig. 1 shows, but we only snip a piece of it to express how to apply the distant supervision to represent a user by linked entities and knowledge types. Details of knowledge assessment can be found in Section 4. Each turk is represented by scored entity instances and probabilistic knowledge types, which are supervised by a distant knowledge graph.

Crowd Formation. Rather than randomly selecting a general set of turks to accomplish tasks, we propose to form a crowd of knowledgeable turks with similarity and diversity. Diversity plays an important role in the process of decision-making, and lack of diversity may result in potential conflict of interests. Besides, when the query is ambiguous, results with diversity can improve the chance of satisfactory answers. However, the challenge is how to measure the similarity and diversity as well as combine them seamlessly into one unified objective function. To be simple, we use sim-div utility maximization as the wise crowd formation objective,

$$\mathcal{T}^* = argmax_{\mathcal{T} \in \mathcal{T}} \sum sim(t, q) div(t, q) \quad (1)$$

which is to select a subset of all turks to maximize the utility function. This unified function takes no parameters to balance the weights of similarity and diversity, which is ideal and easy to use in practice.

The credibility of a group of turks depends not only on the domain knowledge of the people who are involved, but also on whether they can give full play to their professional knowledge and make a wise and fair decision. According to Plachouras [8], taking diversity seriously may also be beneficial for expertise retrieval. Details of wise crowd formation can be found in Section 5. We prove the wise crowd formation a NP-hard problem and propose greedy algorithms to approach.

3.2 System Architecture

The system architecture of our wise crowd find with distant supervision is illustrated in Fig. 2. As the figure shows, both the knowledge assessment and crowd formation are aided by knowledge graph, i.e., distant supervision. Distant supervision can help with the accuracy of matching and guidance of diversity. The most left part is social network users corpus maintenance module, which is based on AsterixDB platform [27]. AsterixDB is a scalable and open-source big data management system with built-in data feed, data storage and query engine. We utilize its data management features into our framework as external social network users data corpus management and extracted turks data storage. The left part in dot-line is built for the knowledge generation and generate the knowledge bi-graph to assess the domain knowledge. The right part in dot-line accounts for the crowd candidate generation and crowd formation. A number of auxiliary indices are constructed over the turks to improve efficiency based on AsterixDB. The distant-supervision knowledge graph is shown at the lower middle part.

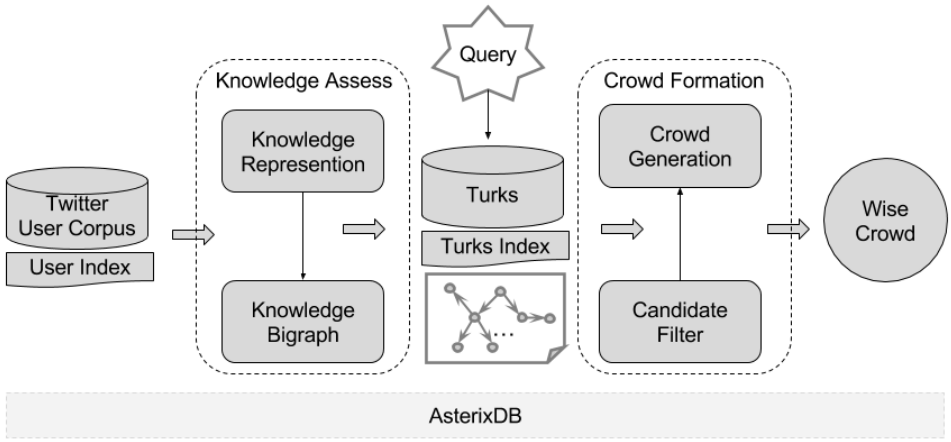


Figure 2: Wise crowd finding system architecture.

To build PIN, we address three key challenges: 1) How to accurately and comprehensively infer an individual user's domain knowledge? 2) How to form a wise crowd of turks with similarity and diversity in domain knowledge on a given query? and 3) How to process millions of users efficiently and scalably?

3.3 Baseline

In this paper, given a query, we aim to find a wise crowd, a set of selected crowdsourcers satisfying both similarity and diversity. It is hard to measure how "wise" a crowd is. However, we can use a naive method named KeyWord Matching (KWM) as the baseline. KWM returns the top turks with highest scores of similarity based on co-occurrences of keywords between query and turks. To be formal, given a query with terms W_q and a set of turks $T = \{t_1, t_2, ..., t_n\}$ with terms W_t^i, the similarity score between the query and each turk is $|W_q \bigcap W_t^i|$. Then turks with highest scores are returned as results. It is not a golden standard for similarity measurement, but a indicator for comparison with PIN to show the performance.

4 PIN-ASSESSOR: DISTANT SUPERVISION BASED KNOWLEDGE ASSESSMENT

We propose distant supervision based knowledge assessment for the social network users to represent their potential domain knowledge. Short texts like tweets vary from traditional documents in their brevity and sparsity, which makes statistical approaches to short texts less effective. Thus, enriching the semantics of short texts using external knowledge, such as Google Knowledge Graph, is essential. We use Twitter in the following paper as the source of potential crowdsourcers. For ease of presentation, we summarize the notations we mainly use in this paper in Table 1.

4.1 Knowledge Representation

Before representing a Twitter user by the Graph, we apply common language processing techniques such as case-folding, stemming, and removal of stop-words to clean the meaningless words in tweets. In addition to the common stop-words, a set of domain-specific stop-words are also filtered out, such emojis in tweets . Then, we segment the tweets into phrases rather than single-word terms by methods in [28]. It is argued that phrases perform better than terms in representing the key-words (-phrases). As tweets are typically short, we consider only uni-grams and bi-grams as phrases. The above strategy produces a set of key phrases for each user, as well as the frequency. That is, for each user u_i in user set $\mathcal{U}$ is associated with a set of key phrases and corresponding frequency, i.e., $\{w : f | f = \frac{|w \in u_i|}{\sum w}\}$.

Given the frequencied phrases of each user, we map each user to a turk by the Graph. As described above, given a query to the Graph, it can return the most relevant scored entities and the corresponding knowledge types. Thus, each phrase can be represented with a set of entities and types, i.e., $\{y, z | y, z \in q(w, G)\}$, where G is the Graph. After the extension, each grassroot user can be represented as a knowledgeable turk by a triple, $< w : f, y : s, z : w >$, i.e., key phrases with frequency, entities with scores and types with weights. We elaborate the definition of entity score and domain weight in the next subsection by the knowledge bi-graph.

4.2 Knowledge Bi-graph

As we can see in Fig. 1, each turk can be expressed as a set of linked entities and knowledge types. The Graph helps us to explore the potential domain knowledge of a turk by the semantic linked entities. Alternatives like Freebase[4] (has been shut down recently but the data dumps are available) and Knowledge Vault [29] (has

[4]https://developers.google.com/freebase/

Notations	Meanings	Notations	Meanings
$\mathcal{U}$	set of social network users	q	query
$\mathcal{W}$	set of phrases for each user	k	top-k number
$f(w,u)$	frequency of phrase w for user$u$	$sim(t,q)$	similarity score of turk t and query q
$\mathcal{T}$	set of crowdsourcing turks	$div(t,q)$	diversity gain of turk t and query q
$\mathcal{Y}$	set of linked entities for each turk	$s(y,t)$	score of turk t at linked entity y
$\mathcal{Z}$	set of knowledge types for each turk	$p(z,t)$	probability of turk t attached to knowledge domain z

Table 1: Table of notations.

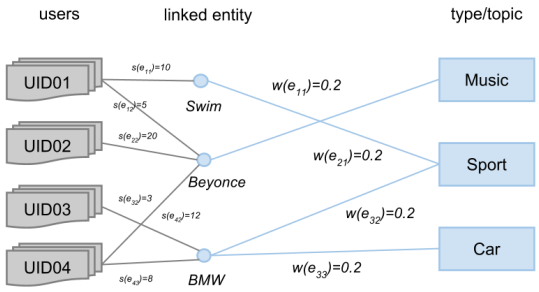


Figure 3: An example of domain knowledge bi-graph.

not been publicly released but the academic paper) also can be used as an external distant supervision.

We represent the matching between social network users and their candidate domain knowledge as a bipartite graph (precisely speaking, two connected bi-graphs), $G_b(\mathcal{U},[\mathcal{Y},\mathcal{Z}],\mathcal{E})$, where $\mathcal{U}$ is the set of users, $[\mathcal{Y},\mathcal{Z}]$ are the sets of entities/types and $\mathcal{E}$ are the links between them in the bi-graph. The type of turk can be interpreted as topic or the domain of knowledge. We exploit availability of the knowledge bi-graph as shown in Fig. 3. We try to get the probabilistic weight $p(y,z)$ that captures the set of relationships between entities and types.

We employ a straightforward similarity function to measure the likelihood of matching an entity to a type in the bi-graph. More specifically, the likelihood of this match is

$$p(y,z) = \frac{|y \in z|}{\sum y} \tag{2}$$

where $|y \in z|$ denotes the number of linked entities y attach to type z. Note that the above matching likelihood can also be measured by any reasonable similarity function. The entity score $s(w,t)$ can be extracted directly from the Graph. We sum up the scores of the same linked entity from different query phrases for each turk.

Therefore, the domain knowledge of each turk can be expressed by linked entities $\mathcal{Y}$ and knowledge types $\mathcal{Z}$. The semantic network of the Graph contains over 570 million objects and more than 70 billion facts about and relationships between different objects that are used to understand the meaning of the keywords queried for the search.

5 PIN-FINDER: WISE CROWD FORMATION BY GREEDY ALGORITHMS

A wise crowd targets not only on similarity of the selected turks, but also on diversity. The objective is to find a group of turks who can collectively perform a task in an effective manner. In this paper, we focus on the similarity of the linked entities and diversity of the knowledge types to form a wise crowd.

HYPOTHESIS (Wise Crowd). *A set of k turks can form a wise crowd given a query if: 1) they maximize the similarity score with the query; 2) they maximize the diversity coverage with the query; 3) number of the crowd is no more than k.*

5.1 Formal Definition

Given:
1) A set of turks $\mathcal{T}$ and each turk t is associated with a set of scored linked entities $\mathcal{Y}$ and a set of probabilistic knowledge types $\mathcal{Z}$, i.e., a) the score of turk t at entity y is $s(y,t)$ and $s(y,t) > 0$, b) the probability of turk t attached to type z is $p(z,t)$ and $\sum_{z\in\ddagger} p(z|,t) = 1$. *For example, turk t_i={piano:15,guitar:35, swim:50, run:10, jump:5; music:0.8, sports:0.2}. In this case, $s(swim,t) = 50$ and $p(music,t) = 0.8$.*
2) A query q which is also associated with a set of scored entities $\mathcal{Y}_q$ and a set of probabilistic types $\mathcal{Z}_q$.
3) A number k.
Goal:
Find a subset of turks $\mathcal{T}^*$ that:
1) $|\mathcal{T}^*| \leq k$
2) sim-div utility maximization:

$$\mathcal{T}^* = argmax_{T\in\mathcal{T}} \sum sim(t,q)div(t,q) \tag{3}$$

where a) $sim(t,q)$ is the similarity score function:

$$sim(t,q) = \sum_{y\in t\cap q} s(y,t)s(y,q) \tag{4}$$

and b) $div(t,q)$ is the diversity gain function:

$$div(t,q) = \sum_{z\in t\cap q} p(z,t)p(z,q)[g(T_{i-1}+t,q) - g(T_{i-1},q)] \tag{5}$$

in which $g(T,q) = \sum_{z\in\mathcal{Z}_q} 1$.

T is any subset of all turks $\mathcal{T}$, $sim(t,q)$ is the sum of similarity scores of each turk t, $div(t,q)$ is the diversity gain after turk k is added to T_{i-1}. To be noted, according to Eq. 5, the diversity gain is non-zero only if the new selected turk t_i cover one more type that has not been covered by previous turks T_{i-1}, which guarantees each new selected turk t_i contributes non-zero diversity gain. The

solution to above equation is $\mathcal{T}^* = (t_1^*, t_2^*, ..., t_k^*)$ that maximizes the sim-div utility of selected turks.

For a general function, the above problem is NP-hard. However, when the function is submodular and monotone, the problem can be cast as finding a maximum-weight basis of a polymatroid, which can be solved greedily and optimally [30]. The sim-div utility function has two notable properties. First, it is parameter-free, which does not require any parameter tuning and therefore should be robust in practice. Second, a greedy method can be applied and therefore it is computationally efficient.

Theorem 5.1. *The wise crowd formation problem is NP-hard.*

Proof. We prove that the wise crowd finding problem is NP-hard by a reduction from the *k Maximum Coverage* (KMC) problem, which is known to be NP-hard.

Recall that an instance of KMC problem (E, S, k) consists of a ground set of items $E = \{s_1, s_2, ..., s_n\}$, a collection of subsets of E, i.e., $S = \{S_1, S_2, ..., S_m\}$ where any set $Si \in S$ satisfies $S_i \subseteq E$, and a number of k. The optimization objective is to select k subsets from S, denoted asS', so that the number of covered items $|\bigcup_{S \in S'} S|$ is maximized.

Our reduction is a weighted version, in which every element s_i has a weight $w(s_i)$. The task is to find a maximum coverage which has maximum weights. The basic version is a special case when all weights are 1. In our instance, the ground truth is the knowledge types of the query $\mathcal{Z}_q = \{z_{q1}, z_{q2}, ..., z_{qn}\}$. The candidate subsets are $\mathcal{Z}_{t_i \in \mathcal{T}}$. The objective is to find a set $\mathcal{T}^*$ that maximizes $\sum sim(t, q)div(t, q)$, where $sim(t, q)$ is the weight and $div(t, q)$ is the weighted number. We show that selecting k turks to form a crowd where the sim-div utility is maximized is equivalent to finding the k best sets that realize maximum coverage with maximum weights. □

Despite the above result shows computing the "wise" crowd is intractable in general, we show that the sim-div utility possesses two good properties, namely monotonicity and submodularity. These properties enable us to develop an algorithm which greedily determines the turks that maximize the sim-div utility and has an approximation ratio of $1 - 1/e$, where e is the base of the natural logarithm [31]. Now, we are ready to prove the monotonicity and submodularity of the sim-div utility, which is shown as follows.

Lemma 5.2. *The sim-div utility given by Eq. 3 is monotone and submodular.*

Proof. Since linear combination of monotone and submodular functions is also monotone and submodular, we only need to show that the term $sim(t, q)div(t, q)$ is monotone and submodular.

We first prove the monotonicity. That is, given two candidate turks with corresponding entity sets and type sets $\mathcal{Y}_{t1}, \mathcal{Y}_{t2}$ and $\mathcal{Z}_{t1}, \mathcal{Z}_{t2}$, if $\mathcal{Y}_{t1} \subseteq \mathcal{Y}_{t2}$, $\mathcal{Z}_{t1} \subseteq \mathcal{Z}_{t2}$, we must have $sim(t_1, q) \leq sim(t_2, q)$, $div(t_1, q) \leq div(t_2, q)$. Consider $\mathcal{Y}_{t1} \subseteq \mathcal{Y}_{t2}$, which indicates $\sum_{y \in t_1 \bigcap q} \subseteq \sum_{y \in t_2 \bigcap q}$, which makes $\sum_{y \in t_1 \cap q} s(y, t_1)s(y, q) \leq \sum_{y \in t_2 \cap q} s(y, t_2)s(y, q)$, i.e., $sim(t_1, q) \leq sim(t_2, q)$. Likewise, $div(t_1, q) \leq div(t_2, q)$ when $\mathcal{Z}_{t1} \subseteq \mathcal{Z}_{t2}$. It is well known if $sim(), div()$ are monotone and $sim() > 0, div() > 0$, then $sim()div()$ is monotone. Hence, the monotonicity is proved.

Next, we prove the submodularity. That is, given two candidate turks with corresponding entity sets $\mathcal{Y}_{t1}, \mathcal{Y}_{t2}$ and a keyword y, if $\mathcal{Y}_{t1} \subseteq \mathcal{Y}_{t2}$, we must have $sim(t_1 \bigcap y, q) - sim(t_1, q) \geq sim(t_2 \bigcap y, q) - sim(t_2, q)$. Consider the monotonicity, we know that $sim(t \bigcap y, q) - sim(t, q) = y$ or $\oslash$, which proves it. The submodularity of $div()$ also can be inferred as the same way without much efforts. Hence, the submodularity is proved. □

5.2 Crowd Formation

Algorithm 1 Wise Crowd Formation (PIN)

Input: $\mathcal{T}$: A ground set of turks
q: A query
k: Size of the crowd
Output: $\mathcal{T}^*$: A subset of $\mathcal{T}$ forming the wise crowd

```
T* ← ∅
/*optimization here*/
for i = 1 to k do
    t* ← max_{t∈T−T*} f(t, q)g(t, q)
    T* ← T* ⋃ t*
end for
return T*
```

The wise crowd formation problem is NP-hard. However, given the properties of monotonicity and submodularity, we propose a greedy algorithm following with optimization tricks in practice. The greedy algorithm is shown as in Algo. 1. Given a query, we compute the sim-div utility score of each turk with the query in each iteration. The turk with highest score is selected to be one of the results and the iteration terminates until k turks are returned. The details are shown in Algo. 1. The run time required for the execution of the algorithm is $O(k \times n)$, where n is the size of turks. However, in practice, we can use some optimization tricks to improve the efficiency.

Algorithm 2 Candidate Filter Optimization (PINOPT)

```
simscorelist=[]
for each turk t in T do
    compute f(t, q), g(t, q)
    if f(t, q) == 0 or g(t, q) == 0 then
        T = T − t
    end if
    if f(t, q)! = 0 then
        simscorelist.add(t)
    end if
end for
```

5.3 Candidate Filter Optimization

Given the query, we first generate possible candidates from the turk database based on keyword co-occurrence of both similarity and diversity, i.e., linked entities and types. That is, turks with zero similarity score or diversity type coverage can be filtered out early. Besides, the similarity score can be computed only once before

Dataset	Size of objects	Size of vocabulary	Total tokens
DBLP	1,712,433	198,527	35,200,971
STOF	2,354,887	37,620	37,138,704
TCKN	973,359	16,166	11,676,869
TEEW	1,578,527	119,692	93,033,27

Table 2: Dataset description.

Parameters	Settings
different datasets	DBLP STOF TCKN **TEEW**
# of objects	**10K** 20K 40K 60K 100K
# of query keywords	1 2 **3** 4 5
# of top-k results	3 5 7 **10** 15

Table 3: Parameters and their settings.

the iteration, which can reduce unnecessary computation. The refinement of the candidate result is carried out in the wise crowd formation. The optimization algorithm is shown in Algo. 2

6 EXPERIMENTS

In this section, we evaluate our proposed approach PIN. We formally evaluate the methods in the context of effectiveness and efficiency. Specifically, we evaluate PIN on time efficiency, phrase similarity and topic diversity based on four different real datasets. All the algorithms are implemented in Python and run on a Windows laptop with an Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz and 8GB memory. We run each experiment three times and report the average results.

6.1 Data Sets

To the best of our knowledge, no available datasets targeting the problem of wise crowd find in social networks exists. Some datasets exist in the context of Enterprise Information Retrieval[5], but they address significantly different content types. We conduct our experiments with four data sets as DBLP, STOF, TCKN and TEEW. DBLP data includes paper information, paper citation, author information and author collaboration[6]. 2,092,356 papers and 8,024,869 citations between them are extracted from academic sources such as DBLP, ACM and CiteSeerX. STOF is a dataset extracted from the Stack Overflow database at 2016-10-13 and contains questions up to 2016-10-12[7]. TCKN is Twitter Foursquare check-ins data with content information across USA except Hawaii and Alaska. The time span is from September 2010 to January 2011. There are 61412 users, 62462 places and 973358 check-ins[8]. TEEW contains 1,578,627 classified tweets used for Twitter setiment analysis[9]. A summary of the four datasets can be found in Table 2. We extract the textual contents and corresponding IDs information as descriptions of social network users. In experiments, we sample different sizes of data and queries from corresponding data sets randomly. Although each data set has objects in millions, we mainly use a small-scale data to run because: 1) Google Knowledge Graph has a strict request limitation when using it as the distant supervision; 2) the limited computation of the experiment laptop. However, we still evaluate our methods in a large-scale dataset TEEW in Appendix A.

[5]http://trec.nist.gov/
[6]https://aminer.org/billboard/aminernetwork
[7]https://github.com/dgrtwo/StackLite
[8]https://sites.google.com/site/dbhongzhi/
[9]http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/

6.2 Evaluation Metrics

It is not as straightforward as it might look like to find an effective evaluation metric. Most conventional metrics try to estimate the likelihood of the test data based on parameters inferred from training or labeling data. However, it is hard to find a standard data set with ground truth. In this paper, we focus on similarity, diversity and efficiency.

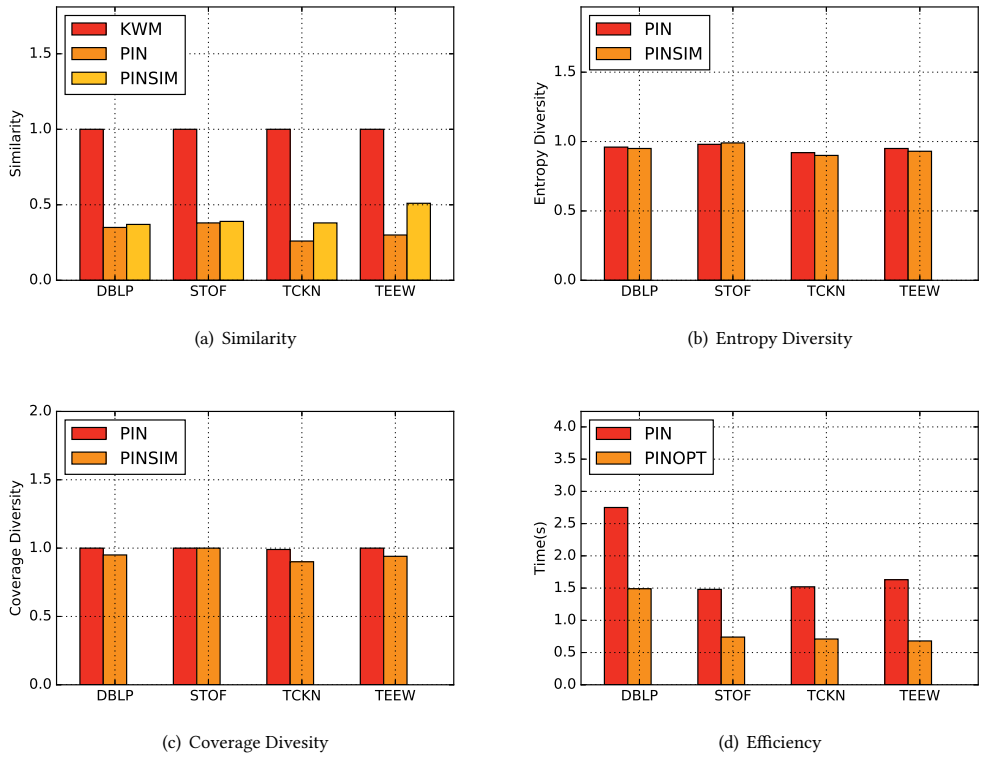
As pointed out, a wise crowd is a set of selected users that satisfy both similarity and diversity. We use a naive method KeyWord Matching (KWM) as the baseline, which is described in Section 3.3. KWM measures the similarity based on co-occurrences of key-words (-phrases) between a query and users. It is an indicator for comparison with PIN on similarity. We name two variants of PIN as PINSIM and PINOPT. PINSIM is the algorithm without diversity constraint in PIN, which can be used to measure similarity only. PINSIM can be regarded as only using distant supervision relatedness to directly generate users correspondences. PINOPT is the optimization of PIN for time efficiency, which applies tricks of filter to improve the computation. In summary, We compare four algorithms, KMW, PIN, PINSIM and PINOPT, i.e., naive method, standard PIN, diversity-free PIN and optimized PIN.

Similarity is measured by the intersection of returned turks from PIN(PINSIM) and KWM. For example, if PIN returns t_1, t_2, t_3 and KWM returns t_2, t_3, t_4 when $k = 3$, then similarity = 2/3, i.e., PIN shares 2 same results with KWM. For diversity, we separate it into entropy diversity and coverage diversity. Entropy diversity measures the entropy of the returned results with query and coverage diversity considers the terms coverage of the returned results for the query. Entropy diversity reflects more information about topic distribution of the returned results than coverage diversity. The diversity measurement is conducted on PIN and PINSIM. Take coverage diversity for example. If PIN returns results covering knowledge types z_1, z_2, z_3, z_4, z_5 and the query covers z_2, z_4, z_6, then the coverage diversity is 2/3, i.e., the covered knowledge types by all results compared to the query. Entropy diversity is similar but measured by entropy. At last, we measure efficiency on PIN and PINOPT. Besides, we measure these metrics with different datasets, varying data size, varying number of query keywords and varying number of returned results. The settings are in Table 3 and the defaults are in bold.

6.3 Experimental Results

The experimental results are explained as follows and additional experiments can be found in appendices.

1) Different datasets. We measure the similarity, diversity and efficiency on four different datasets from a general view. As shown in Fig. 4 (a), PIN and PINSIM share about 30-50% same results



(a) Similarity

(b) Entropy Diversity

(c) Coverage Divesity

(d) Efficiency

Figure 4: Different Datasets.

with KWM. To be noted, KWM is not the standard, but just a indicator to compare with. From the figure, we can see that if we take diversity into consideration (PIN), the similarity becomes lower. This is because we give priority not only to similarity but also to diversity, which makes high similarity score turks with little diversity contribution may fail to be selected. Fig. 4 (b) (c) further shows how the diversity works in forming a wise crowd. However, without diversity into consideration explicitly, the formed crowd still keeps a certain diversity level, which can explain why most similarity-based search strategies are successful in practice. In Fig. 4 (d), we can see that the optimization tricks of filter accelerates the computation greatly. The time of DBLP is more than that of other datasets, which mainly results from the long average tokens in DBLP dataset, which has about 20 tokens for each object as shown in Table 2.

2) Varying data size. We examine the scalability by varying the size of TEEW from 10K to 100K. As shown in Fig. 5, similarity and diversity have no apparent increase or decrease when the data size grows. This indicates the methods can keep robust performance on different data sizes. However, the time increases as data size grows, which is reasonable. Still, PINOPT shows its advantage over un-optimized PIN. The abnormal point in Fig. 5 (a) that PIN shows slightly higher similarity than PINSIM maybe be due to indicator KWM is not always the best to compare with.

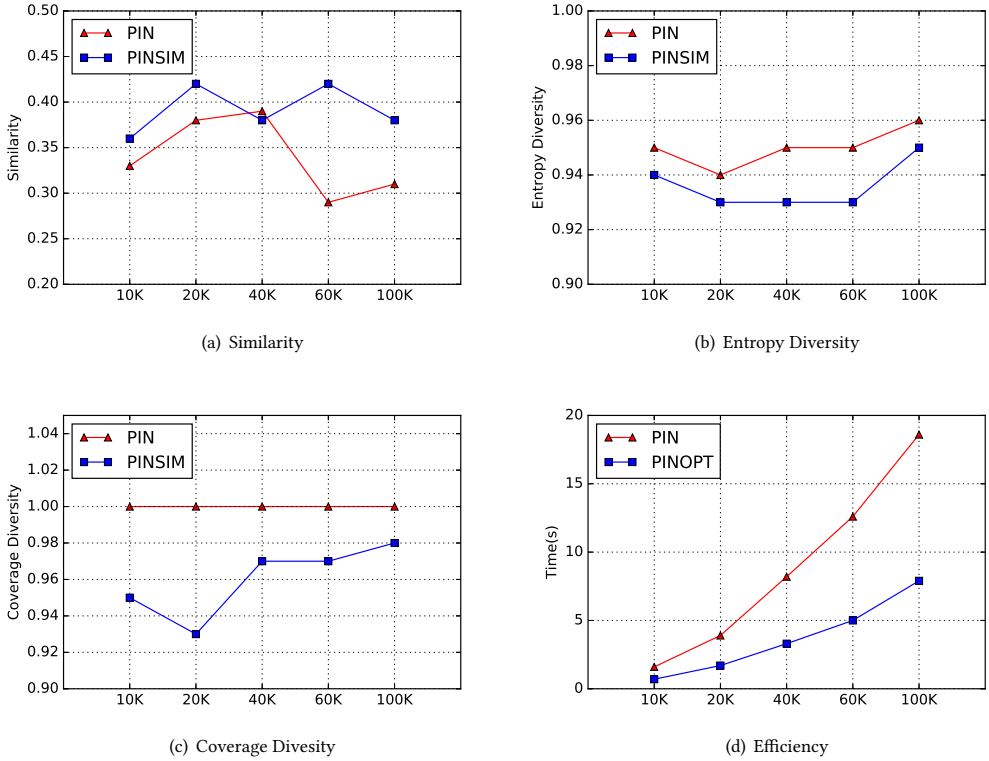
3) Varying number of query keywords. We vary the number of query keywords from 1 to 5 to evaluate the efficiency. As the number of query keywords increases, it takes longer time to get the results as shown in Fig. 6 (a). This is mainly because it takes more time to compute the similarity and diversity when a query contains more keywords.

4) Varying number of returned results. We vary the returned results number i.e., k of top-k, from 3 to 15 to evaluate the efficiency as results are shown as Fig. 6 (b). It is obvious that it takes more time to get more results given same other conditions. And we can see the time of PINOPT increases gradually .

7 CONCLUSION

In this paper, we design and implement a scalable framework to find potential users from existing social networks for crowdsourcing. We propose distant supervision to assess the domain knowledge and formulate the wise crowd formation as NP-hard problem with greedy algorithms presented. Experiments show the performance of our solution in aspects of effectiveness and efficiency. A brief demonstration can be accessed at http://radon.ics.uci.edu:9118.

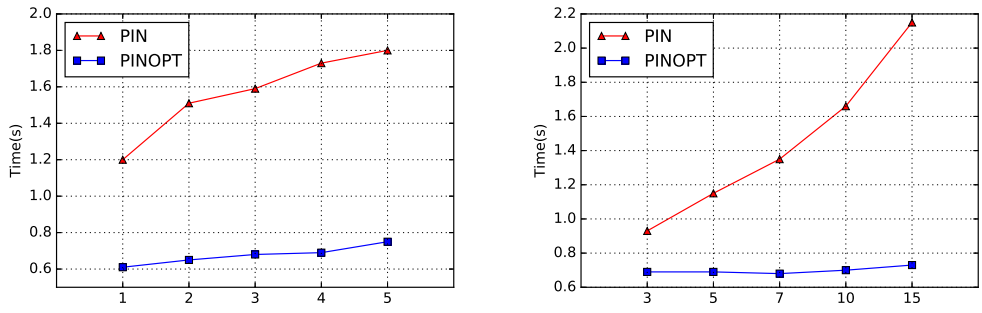
The methods can be used on more than potential users finding in crowdsourcing. More prospective future can be exampled as first batch of targeted users, ad promotion and interest group mining. While we provide a scalable framework to find potential users from grassroots, it still needs researches whether these potential users are easily to be transformed to join and contribute in the crowdsourcing.



(a) Similarity (b) Entropy Diversity

(c) Coverage Divesity (d) Efficiency

Figure 5: Varying Data Size.



(a) Query Keywords (b) Returned Results

Figure 6: Varying Parameters.

Acknowledgment

This work is supported by National Science Foundation of China (No.61532021), National Basic Research Program of China (973) (No.2014CB340403), and National High Technology Research and Development Program of China (863) (No.2014AA015204).

REFERENCES

[1] M. Vukovic, S. Kumara, and O. Greenshpan, "Ubiquitous crowdsourcing," in *Ubicomp 2010*. ACM, 2010, pp. 523–526.

[2] J. Ren, Y. Zhang, K. Zhang, and X. Shen, "Exploiting mobile crowdsourcing for pervasive cloud services: challenges and solutions," *Communications Magazine, IEEE*, vol. 53, no. 3, pp. 98–105, 2015.

[3] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment framework for participatory sensing data collections," in *Pervasive Computing*. Springer, 2010, pp. 138–155.

[4] H.-L. Yang and C.-Y. Lai, "Motivations of wikipedia content contributors," *Computers in Human Behavior*, vol. 26, no. 6, pp. 1377–1383, 2010.

[5] A. Burnap, Y. Ren, R. Gerth, G. Papazoglou, R. Gonzalez, and P. Y. Papalambros, "When crowdsourcing fails: A study of expertise on crowdsourced design evaluation," *Journal of Mechanical Design*, vol. 137, no. 3, p. 031101, 2015.

[6] P. G. Ipeirotis and P. K. Paritosh, "Managing crowdsourced human computation: a tutorial," in *WWW 2011*. ACM, 2011, pp. 287–288.

[7] H. Yin, B. Cui, and Y. Huang, "Finding a wise group of experts in social networks," in *ADMA 2011*, 2011, pp. 381–394.
[8] V. Plachouras, "Diversity in expert search," in *Workshop on Diversity in Document Retrieval*, 2011, pp. 63–67.
[9] H. Gao, C. H. Liu, W. Wang, J. Zhao, Z. Song, X. Su, J. Crowcroft, and K. K. Leung, "A survey of incentive mechanisms for participatory sensing," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 2, pp. 918–943, 2015.
[10] K. Balog, L. Azzopardi, and M. de Rijke, "Formal models for expert finding in enterprise corpora," in *SIGIR 2006*, 2006, pp. 43–50.
[11] J. Tang, L. Yao, D. Zhang, and J. Zhang, "A combination approach to web user profiling," *TKDD*, vol. 5, no. 1, p. 2, 2010.
[12] A. Daud, J. Li, L. Zhou, and F. Muhammad, "Temporal expert finding through generalized time topic modeling," *Knowledge-Based Systems*, vol. 23, no. 6, pp. 615–625, 2010.
[13] H. Deng, I. King, and M. R. Lyu, "Enhanced models for expertise retrieval using community-aware strategies," *IEEE Trans. Systems, Man, and Cybernetics, Part B*, vol. 42, no. 1, pp. 93–106, 2012.
[14] I. Guy, U. Avraham, D. Carmel, S. Ur, M. Jacovi, and I. Ronen, "Mining expertise and interests from social media," in *WWW 2013*, 2013, pp. 515–526.
[15] X. Quan, C. Kit, Y. Ge, and S. J. Pan, "Short and sparse text topic modeling via self-aggregation," in *IJCAI 2015*, 2015, pp. 2270–2276.
[16] E. Smirnova, "A model for expert finding in social networks," in *SIGIR 2011*, 2011, pp. 1191–1192.
[17] J. Zhang, J. Tang, and J. Li, "Expert finding in a social network," in *DASFAA 2007*, 2007, pp. 1066–1069.
[18] P. Serdyukov, H. Rode, and D. Hiemstra, "Modeling multi-step relevance propagation for expert finding," in *CIKM 2008*, 2008, pp. 1133–1142.
[19] A. Bozzon, M. Brambilla, S. Ceri, M. Silvestri, and G. Vesci, "Choosing the right crowd: expert finding in social networks," in *EDBT 2013*, 2013, pp. 637–648.
[20] P. G. Ipeirotis and E. Gabrilovich, "Quizz: targeted crowdsourcing with a billion (potential) users," in *WWW 2014*, 2014, pp. 143–154.
[21] S. Ghosh, N. K. Sharma, F. Benevenuto, N. Ganguly, and P. K. Gummadi, "Cognos: crowdsourcing search for topic experts in microblogs," in *SIGIR 2012*, 2012, pp. 575–590.
[22] J. Kang and K. Lerman, "Leveraging user diversity to harvest knowledge on the social web," in *PASSAT/SocialCom 2011*, 2011, pp. 215–222.
[23] H. Su, J. Tang, and W. Hong, "Learning to diversify expert finding with subtopics," in *PAKDD 2012*, 2012, pp. 330–341.
[24] L. Robert and D. M. Romero, "Crowd size, diversity and performance," in *CHI 2015*, 2015, pp. 1379–1382.
[25] C.-J. Ho and J. W. Vaughan, "Online task assignment in crowdsourcing markets." in *AAAI 2012*, vol. 12, 2012, pp. 45–51.
[26] W. X. Zhao, J. Jiang, J. Weng, J. He, E. Lim, H. Yan, and X. Li, "Comparing twitter and traditional media using topic models," in *ECIR 2011*, 2011, pp. 338–349.
[27] S. Alsubaiee, Y. Altowim, H. Altwaijry, A. Behm, V. Borkar, Y. Bu, M. Carey, I. Cetindil, M. Cheelangi, K. Faraaz *et al.*, "Asterixdb: A scalable, open source bdms," *Proceedings of the VLDB Endowment*, vol. 7, no. 14, pp. 1905–1916, 2014.
[28] A. El-Kishky, Y. Song, C. Wang, C. R. Voss, and J. Han, "Scalable topical phrase mining from text corpora," *PVLDB*, vol. 8, no. 3, pp. 305–316, 2014.
[29] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in *SIGKDD 2014*. ACM, 2014, pp. 601–610.
[30] J. Edmonds, "Submodular functions, matroids, and certain polyhedra," *Combinatorial structures and their applications*, pp. 69–87, 1970.
[31] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions - I," *Math. Program.*, vol. 14, no. 1, pp. 265–294, 1978.

8 APPENDIX A: PIN ON LARGE-SCALE DATASETS

We estimate the possible cost when applying our method to a large-scale social network corpus (TEEW) as we use term "millions" in title. In Fig. 7, SML data size is 100K, diversity is coverage diversity, Sim, Div, Eff is based on PIN, SimSIM, DivSIM are based on PINSIM and EffOPT is based on PINOPT. We integrate all the results in one figure without confusion. The y-label is showed in percentage except for efficiency measurement. As we can see from the figure, the most apparent effect of large scale data is on efficiency. It takes about 10x times without optimization, which proves the tricks can filter most zero-similarity and non-coverage turks in practice.

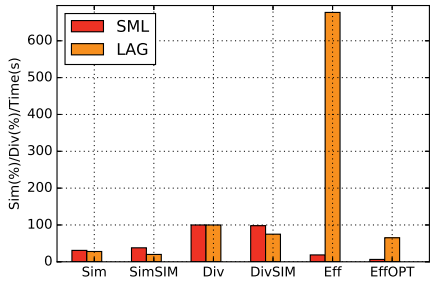
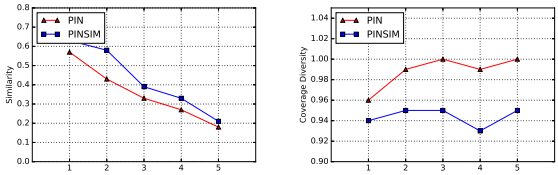
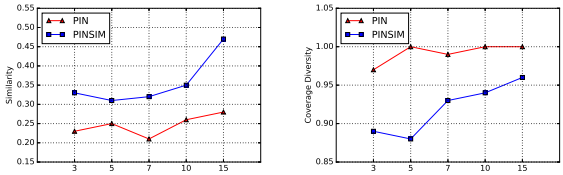


Figure 7: Large-scale Data.



(a) Similarity (b) Coverage Diversity

Figure 8: Query Keywords.



(a) Similarity (b) Coverage Diversity

Figure 9: Returned Results.

9 APPENDIX B: MORE ADDITIONAL EXPERIMENTS

The parameters of number of query keywords and number of returned results mainly affect the efficiency, but we still post the effects on similarity and diversity in Fig. 8 and Fig. 9. The only thing to be mentioned is that the similarity shows apparent decrease when the query keywords increase. There are two possible explanations: 1) the decrease is shown compared with KWM, which is just an indicator; 2) increase of query keywords means more information in the query and diversity may dominate in the process when finding the wise crowd.

Further, at http://radon.ics.uci.edu:9110/pin a brief demonstration can be accessed. We can try some queries and get the wise crowd and statistics. Prompts are given in the placeholders to help with the use. If the site is temporally unavailable, please email ideamaxwu@gmail.com to restart it or further solution.