



# Retrieving Skill-Based Teams from Collaboration Networks

Radin Hamidi Rad  
radin@ryerson.ca  
Ryerson University, Canada

Ebrahim Bagheri  
bagheri@ryerson.ca  
Ryerson University, Canada

Mehdi Kargar  
kargar@ryerson.ca  
Ryerson University, Canada

Divesh Srivastava  
divesh@research.att.com  
AT&T Labs-Research, USA

Jaroslav Szlichta  
jarek@ontariotechu.ca  
Ontario Tech University, Canada

## ABSTRACT

Given a set of required skills, the objective of the team formation problem is to form a team of experts that cover the required skills. Most existing approaches are based on graph methods, such as minimum-cost spanning trees. These approaches, due to their limited view of the network, fail to capture complex interactions among experts and are computationally intractable. More recent approaches adopt neural architectures to learn a mapping between the skills and experts space. While they are more effective, these techniques face two main limitations: (1) they consider a fixed representation for both skills and experts, and (2) they overlook the significant amount of past collaboration network information. We learn dense representations for skills and experts based on previous collaborations and bootstrap the training process through transfer learning. We also propose to fine-tune the representation of skills and experts while learning the mapping function. Our experiments over the DBLP dataset verify that our proposed architecture is able to outperform the state-of-the-art graph and neural methods over both ranking and quality metrics.

## CCS CONCEPTS

• **Information systems** → *Retrieval models and ranking*; **Expert search**; • **Computing methodologies** → *Search methodologies*.

## KEYWORDS

Team Formation, Expert Networks, Task Assignment

### ACM Reference Format:

Radin Hamidi Rad, Ebrahim Bagheri, Mehdi Kargar, Divesh Srivastava, and Jaroslav Szlichta. 2021. Retrieving Skill-Based Teams from Collaboration Networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*, July 11–15, 2021, Virtual Event, Canada. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3404835.3463105>

## 1 INTRODUCTION

As the nature of work is becoming increasingly collaborative and interdisciplinary, the need for teamwork between multiple experts

is now of paramount importance. For this reason, the problem of team formation, which is focused on bringing together groups of experts that can collectively address a set of requirements, receives attention. The objective of the *team formation problem* is to find a group of experts who can effectively work together and cover a set of skills that are required for completing a task. For instance, when working on a website that includes an intelligent conversation agent, one would need to assemble experts who have expertise in ‘web development’, ‘user experience design’, ‘natural language processing’, and ‘information retrieval’.

Traditionally, the problem of team formation has been viewed as one of mining a subgraph from a *collaboration network* representing experts, their skills and their past teamwork. Specifically, Sozio et al. [19] offer an encompassing definition for the problem of team formation. They define an appropriate team, as one represented by an induced subgraph that maximizes a monotone optimization function under a set of constraints. Depending on the application area, different authors have provided concrete implementations of the monotone optimization function. Lappas et al. [13] make use of the minimum-cost spanning tree that satisfies the constraints (e.g., covering all required skills). More recent work by Kargar et al. [9] and Bryson et al. [2] focus on minimizing the sum of the weights of the induced subgraph. However, the major limiting factor for the class of solutions that identify subgraphs as teams is that they are all computationally intractable since subgraph optimization techniques have been shown to be a reduced version of the Steiner-tree problem, which is NP-hard [10].

Given the computational intractability of the methods that identify subgraphs as teams, researchers have also considered using neural architectures to identify teams. The major objective of these works is to learn an efficient mapping from the space of skills onto the space of experts, so that expert teams can be formed effectively in real-time. To the best of our knowledge, Sapienza et al. [17] were the first to employ an autoencoder architecture to identify experts that can learn from other team members. However, given the fact that the distribution of skills over experts and teams is quite sparse, a non-variational autoencoder architecture may be prone to overfitting. To address the issue with sparsity of skill distributions, Rad et al. [16] built on the foundations of the work by Sapienza et al. and proposed a variational Bayesian neural architecture. The authors showed that a variational architecture can lead to a more effective treatment of sparsity, hence, producing better quality teams.

However, both of these approaches face two main drawbacks. (1) They assume that the representations of teams and skills are predetermined and as such their objective is to learn a mapping between these fixed representations. Neither approaches attempt to

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
SIGIR '21, July 11–15, 2021, Virtual Event, Canada

© 2021 Association for Computing Machinery.  
ACM ISBN 978-1-4503-8037-9/21/07...\$15.00  
<https://doi.org/10.1145/3404835.3463105>

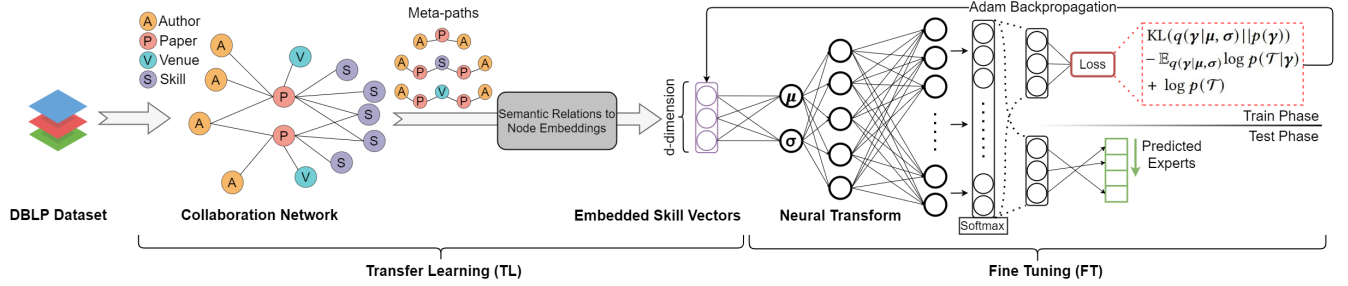


Figure 1: Illustration of our model architecture with transfer learning and fine-tuning components.

co-train the skill and expert representations during the process of learning the mapping between the skill and expert spaces. (2) Given the complexities associated with processing large graphical structures, both Rad et al. and Sapienza et al. overlook the rich information that can be mined from the associated collaboration network.

We focus on advancing the state-of-the-art in neural network methods for team formation by addressing the above two limitations. As such, our work offers the following two contributions. (1) We learn dense representations for the nodes in the collaboration network, e.g., skills and experts, and employ it to bootstrap our learning process. We refer to this as a *transfer learning* process. (2) We design our neural architecture in a way that the representations of skills and experts are learned in tandem with the mapping between the skill and expert spaces. We refer to this process as a *fine-tuning* process as the initial representations of skills and experts are initially trained using the transfer learning process based on the collaboration network. We show that the transfer learning and fine-tuning processes provide significant performance and accuracy improvements over the state-of-the-art neural architectures for team formation [16, 17], as well as, graphs methods [2, 9, 13].

## 2 METHODOLOGY

### 2.1 Architecture Overview

The main objective of team formation is to form a team of experts that have a given set of skills. For each team, let  $\mathcal{S} = \{\mathcal{S}_i\}$  represent the skill set and  $\mathcal{E} = \{\mathcal{E}_j\}$  denote experts. Each team is represented as  $(\mathbf{s}, \mathbf{e})$ , where  $\mathbf{e} \subseteq \mathcal{E}; \mathbf{e} \neq \emptyset$  and  $\mathbf{s} \subseteq \mathcal{S}; \mathbf{s} \neq \emptyset$ . We let  $\mathcal{T} = \{(s_i, e_i)\}_{i=1}^N$  be a dataset consisting of  $N$  pairs of skills and experts.

Our proposed model learns a *transform* function  $f : (\mathcal{S}) \rightarrow (\mathcal{E})$  that forms a team of experts for a set of skills. The architecture is illustrated in Fig. 1. Our system architecture consists of two main components: (1) a *transfer learning* (TF) component that learns initial embeddings of the skill and expert nodes used to bootstrap the training of the transform function; and, (2) the *fine-tuning* (FT) process, which trains  $f : (\mathcal{S}) \rightarrow (\mathcal{E})$  and fine-tunes the skill and expert representations in tandem to predict potential expert candidates for forming a team for the given set of skills.

### 2.2 Transfer Learning

We hypothesize that the past collaboration of experts and their prior engagement with different skills can positively impact how teams are formed in the future. We utilize the *transfer learning* component to learn representations for experts and skills based on how they interacted with each other within a graph collaboration network. A collaboration network can be represented by a heterogeneous graph  $G(V, E, T)$ , where  $V$  is the set of nodes, each of which is associated with a type defined by the mapping function  $\phi(v) : V \rightarrow T_V$  and  $E$  is a set of edges connecting the heterogeneous nodes. For

instance, for the DBLP dataset that is used in our experiments,  $T_V$  consists of paper, author, venue and skill node types.

We use a heterogeneous skip-gram model to capture the heterogeneous neighborhood of each node in the graph. Some existing heterogeneous graph embedding techniques apply random walk over the network to build node sequences required by the skip-gram model to learn node representations [3, 18]. However, Sun et al. [21] showed that the random walk strategy can introduce bias towards node types that are more dominant in the network. Thus, we perform targeted random walk by using a meta-path scheme as suggested by Fard et al. [5] and Dong et al. [4]. This allows us to capture the semantic and structural correlations between different targeted types of nodes and their relationships, which are important in the context of team formation. We take advantage of the meta-paths [6, 21, 22] deployed for the DBLP dataset (c.f. Figure 1), which capture semantic relations including co-authorship and authors who publish in the same venues.

Given the collaboration network, our meta-path scheme is formulated as follows:  $\mathcal{P} : V_1 \xrightarrow{R_1} V_2 \xrightarrow{R_2} \dots \xrightarrow{R_t} V_{t+1} \dots \xrightarrow{R_{l-1}} V_l$ , where  $R = R_1 \circ R_2 \circ \dots \circ R_{l-1}$  is synthesized from relations between node types  $V_1$  and  $V_l$  [20]. The probability of moving to the next node for a random walker is defined as follows:

$$p(v^{i+1} | v_t^i, \mathcal{P}) = \begin{cases} \frac{1}{|N_{t+1}(v_t^i)|} & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) = t+1 \\ 0 & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) \neq t+1 \\ 0 & (v^{i+1}, v_t^i) \notin E \end{cases} \quad (1)$$

where  $v_t^i \in V_t$ . Here,  $N_{t+1}(v_t^i)$  specifies the next node's type ( $V_{t+1}$ ) based on the given meta-path  $\mathcal{P}$ . Since meta-path schemes are designed in a symmetric way, its first node type  $V_1$  is the same as the last node in the meta-path  $V_l$  [20, 21],

$$p(v^{i+1} | v_t^i) = p(v^{i+1} | v_1^i), \text{ if } t = l \quad (2)$$

Hence, a random walker is able to recursively walk through a target node neighborhood and capture deeper semantic relations. The targeted random walk produces a set of node sequences that are used by a skip-gram model with sensitivity to node types to learn node representations:

$$X \in \mathbb{R}^{|V| \times d}, d \ll |V| \quad (3)$$

During the embedding training, we maximize the probability of having the heterogeneous context  $N_t(v)$ ,  $t \in T_V$  for a node  $v$ :

$$\arg \max_{\theta} \sum_{v \in V} \sum_{t \in T_V} \sum_{c_t \in N_t(v)} \log p(c_t | v; \theta) \quad (4)$$

where  $N_t(v)$  refers to the neighborhood of node  $v$  and nodes of type  $t$ . The mentioned probability function, i.e.,  $p(c_t | v; \theta)$ , can be a softmax function [1, 7]. In order to reach an optimized representation, we utilized negative sampling [15]. Given a negative sampling

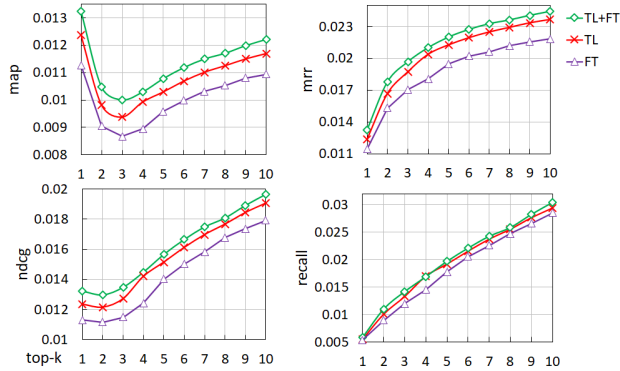


Figure 2: Performance of the variants of our approach.

size  $M$ ,  $p(c_t | v; \theta)$  in Eq. 4 is updated as:

$$\log \sigma(X_{c_t} \cdot X_v) + \sum_{m=1}^M \mathbb{E}_{u^m \sim P(u)} [\log \sigma(-X_u^m \cdot X_v)] \quad (5)$$

where  $\sigma(x) = \frac{1}{1+e^{-x}}$ . We select  $M$  negative samples drawn from a random distribution  $P(u)$  (homogeneously chosen based on the frequency of node types). The skill and expert representations learned from the collaboration network are used to bootstrap the fine tuning component, which will further tune these representations in tandem with the training of the transform function.

### 2.3 Fine Tuning

Next, we devise the component to learn a transform function from the skills space to the experts space in tandem with fine-tuning the embeddings from the transfer learning component. We first use the embedding representations from the transfer learning component to initialize our embedding vectors (see Fig. 1). We then train our neural network using the generated vectors whilst the embedding vectors are fine-tuned.

$$l_0 = \rho(\overline{X_s}) \quad (6)$$

$$l_i = F_i(W^i l^{i-1}) \quad (7)$$

$$O = F_m(W^m l^{m-1}) \quad (8)$$

where  $m$  is the number of hidden layers and  $\overline{X_s}$  is the input layer synthesized from the transferred skill embeddings.  $\rho(\cdot)$  calculates  $\mathcal{N}(\mu, \sigma^2)$  for the input layer. For each of the hidden layers  $l_i, i \in \{1, m\}$ ,  $F_i$  denotes their activation function and  $W^i \in \mathbb{R}^{i-1 \times i}$  is the weight matrix. Take note that in our formulation, bias term is included in weight matrices. We employ the commonly used softmax function for the activation function of the output layer  $F_m$ . We fine-tune the embeddings during the train phase using Adam [11] for back propagation.  $\Theta = \{X_s\}_{s \in S}, \{W^{(i)}\}_{i=0}^m$  represents all trainable model parameters. For the test phase, each neuron of the output layer holds probability for each expert to be part of the team.

Next, we define our loss function based on the characteristic of the collaboration network, where the connection between the nodes is sparse. The neural architectures based on a generative approach are more stable and efficient, than the discriminative ones [8]. In a generative approach, the distribution of skills is considered instead of their deterministic values. Our goal is to fine-tune a transform function, with the above neural architecture, with parameter  $\gamma$  of the conditional probability  $p(e | s, \gamma)$ . We leverage the Bayes rule to infer the posterior distribution after calculating the prior distribution of  $\gamma$ . However, computing the posterior distribution  $p(\gamma | \mathcal{T}) = p(\gamma)p(\mathcal{T} | \gamma)/p(\mathcal{T})$  is difficult, because  $p(\mathcal{T}) = \int p(\mathcal{T}, \gamma)d\gamma$  is a computationally intractable integral. Thus,

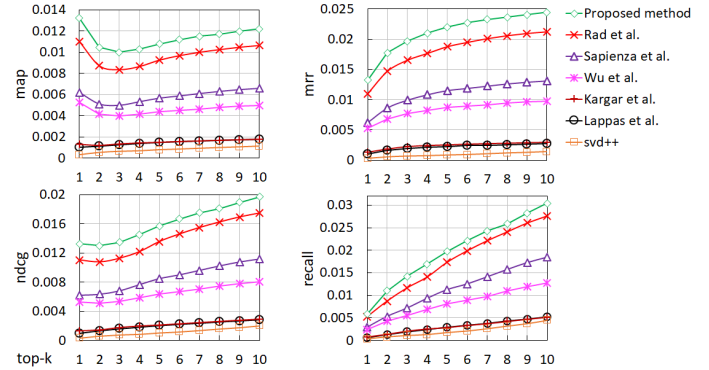


Figure 3: Performance of our approach vs baselines.

we use variational inference, an efficient approximation approach to tackle this problem by calculating  $q_\phi(\gamma)$  to approximate  $p(\gamma | \mathcal{T})$ . We minimize the difference between  $q_\phi(\gamma)$  and  $p(\gamma | \mathcal{T})$  using Kullback-Leibler divergence as follows:

$$\text{KL}(q(\gamma|\mu, \sigma)||p(\gamma|\mathcal{T})) = \int q(\gamma|\mu, \sigma) \log \left[ \frac{q(\gamma|\mu, \sigma)}{p(\gamma|\mathcal{T})} \right] d\gamma \quad (9)$$

$$= \mathbb{E}_{q(\gamma|\mu, \sigma)} \log \left[ \frac{q(\gamma|\mu, \sigma)}{p(\mathcal{T}|\gamma)p(\gamma)} \right] \quad (10)$$

$$= \text{KL}(q(\gamma|\mu, \sigma)||p(\gamma)) - \mathbb{E}_{q(\gamma|\mu, \sigma)} \log p(\mathcal{T}|\gamma) + \log p(\mathcal{T}) \quad (11)$$

We train our network with the loss function mentioned in Eq. 11.

## 3 EXPERIMENTS

### 3.1 Settings

**Datasets.** The common dataset that has been widely used in the literature for evaluating team formation methods has been based on the DBLP publication repository [14, 16, 24]. The authors of each paper are considered to represent a team, and the specializations required for the paper are its skill sets. As suggested by [9], after stop word removal and stemming, we identify the top-2,000 unigram, bigram and trigrams of the phrases with the highest tf-idf scores from the paper titles to represent the set of skills. We construct the heterogeneous collaboration network from the DBLP dataset by considering the authors, papers, skills, and venues as the nodes of the network. The edges of the collaboration network represent associations between author-paper, paper-skill and paper-venue node pairs. Our dataset, its statistics and the code for our work is publicly available<sup>1</sup>.

**Metrics.** We evaluate the effectiveness of our approach from two complementary perspectives. In the first perspective, we adopt widely used ranking metrics, namely mean average precision (map), mean reciprocal rank (mrr), normalized discounted cumulative gain (ndcg), and recall. In the second perspective, we measure the quality of the recommended teams based on two metrics, namely skill coverage (sk), and team comparability (tk). The skill coverage metric measures the percentage of the required skills that are actually covered by the proposed team. Ideally, the proposed team should fully cover all of the required skills. On the other hand, team comparability shows to what extent the proposed team is comparable to the actual set of authors. We compute the difference between the average h-index of the proposed team and that of the actual team. The

<sup>1</sup> <https://github.com/radinhamidi/Retrieving-Skill-Based-Teams-from-Collaboration-Networks>

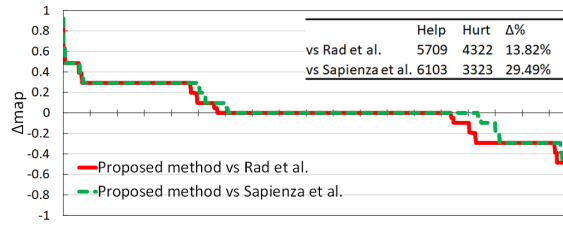


Figure 4: Help/Hurt results on per-team basis.

closer the proposed team is to the actual team, the lower the tk metric will be. All reported results are based on 10-fold cross validation.

**Baselines.** We select our baselines based on the classification mentioned in Sec. 1: **Graph Methods:** Kargar et al. [9] have argued that team formation is a special case of keyword search over graphs and provide optimization-based solutions to compute node and edge weights in the graph to evaluate possible ranking strategies. Lapas et al. [14] pioneered the idea of team formation on graphs and offered heuristic solutions to find subgraphs based on subtrees with minimal diameter. **Neural methods:** Sapienza et al. [17] have suggested that an autoencoder architecture can be used determine teams that can maximize skill transfer between experts. Rad et al. [16] have recently proposed a variational Bayesian neural network that maps skills to experts through a single hidden layer. **CF methods:** In addition, team formation can also be viewed as a collaborative filtering (CF methods) task where team members are suggested based on a set of input skills. We adopt Wu et al. [23] who propose the recurrent recommender network that uses a Long Short-Term Memory (LSTM) autoregressive model for capturing the relation between items and users. Finally, we also include the widely known svd++ [12] method used as a recommender. We note when required all method hyperparameters were tuned or selected based on the authors' suggestion.

### 3.2 Findings

**Ranking.** We first analyze the impact of the variants of our proposed approach on the ranking metrics. The results are shown in Fig. 2. We make several important observations: (1) overall, the transfer learning (TL) approach has a stronger impact on the performance of our approach compared to the fine-tuning (FT) strategy by at least 6% on map, ndcg and mrr. However, the performance is similar over recall. (2) despite the better performance of transfer learning, when integrated with fine-tuning strategy (TL+FT), the overall performance of our approach increases between 3.5% and 5.5% on map, ndcg and mrr. This shows that although the fine-tuning strategy is not as effective as transfer learning, it does have complementary and synergistic impact on performance.

Next, we compare the best variant of our approach, against the baselines in Fig. 3. We note important findings: (1) our approach consistently outperforms all other baselines on all four ranking metrics. The degrees of improvement are consistent across all top- $k$  ranks and are 22%, 12.6%, 15%, and 10.5% over the best baseline (Rad et al.) for the map, ndcg, mrr and recall, respectively. (2) Neural network-based baselines offer a better performance compared to graph-based baselines, even the-state-of-the-art method proposed by Kargar et al. This poorer performance for graph-based methods can be attributed to the NP-hard nature of team formation over graphs, which leads to sub-optimal greedy solutions. (3) When comparing to the neural baselines, the differentiating aspect of our

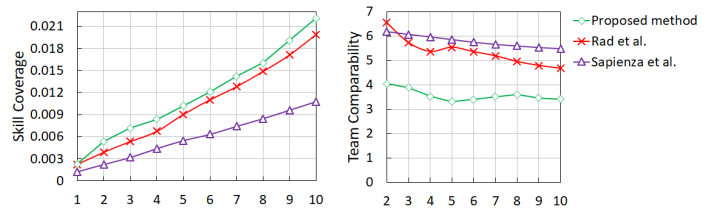


Figure 5: Qualitative measures of performance.

work, from Sapienza et al. and Rad et al., is that we allow for our model to learn association between skills and experts, through transfer learning and fine-tuning of embeddings, whereas earlier work resort to only learning a mapping between the skill and expert spaces. For this reason, even when compared to the weakest variant of our approach (FT in Fig. 2), the performance of our approach is stronger than the best baseline on all ranking metrics.

Furthermore, we plot the degree to which map was helped or hurt by our approach compared to the two best neural baselines on a per-team basis in Fig. 4. The figure shows that 56.9% and 64.7% of the map values were improved when compared to Rad et al. and Sapienza et al., respectively. This is equivalent to 13.8% and 29.4% more teams being helped compared to the baselines pointing to a consistent improvement across a large number of teams.

**Quality.** While the ranking metrics evaluated whether the original authors of a paper were retrieved by the team formation methods, the quality metrics are focused on two additional aspects of teams: (1) skill coverage (sk), which evaluates whether the proposed team actually covers the required set of skills that were requested, regardless of whether the original authors were retrieved or not, and (2) team comparability (tk) that measures how similar the retrieved authors are in terms of their h-index. The idea behind sk and tk is that if the original set of authors is not retrieved for forming a team, the team would need to ideally have the same set of skills as the original authors, i.e., sk, and have the same stature as them, i.e., tk. As noted earlier, higher sk and lower tk values are desirable.

Fig. 5 reports the results of comparison with the two strongest baselines. We make two important observations: (1) our method performs consistently better in terms of skill coverage compared to both baselines. The improvement on sk is at least 11% over the stronger baseline, i.e., Rad et al. over the different team sizes (top- $k$ ). This is an indication that our approach is able to find teams that cover a larger set of the required skills. (2) From the tk perspective, our approach finds teams that have a lower difference to the original authors in terms of their average team h-index compared to the baselines, which is at least 37% better than the best baseline. We believe this is due to how the information about author and skill relations are transferred from the collaboration network and then those representations are fine-tuned in our approach. This leads to the retrieval of similar profile team members even when the original team members are not retrieved.

## 4 CONCLUDING REMARKS

We provide advancements over existing neural architectures designed for the team formation task through (1) a transfer learning process and (2) fine-tuning of skill and expert embedding representations. Through our experiments on the DBLP dataset, we illustrate that our proposed approach shows stronger performance compared to the state-of-the-art over both ranking and quality metrics.

## REFERENCES

- [1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.
- [2] Spencer Bryson, Heidar Davoudi, Lukasz Golab, Mehdi Kargar, Yuliya Lytvyn, Piotr Mierzejewski, Jaroslaw Szlichta, and Morteza Zihayat. 2020. Robust keyword search in large attributed graphs. *Information Retrieval Journal* 23, 5 (2020), 502–524.
- [3] Lu Chen, Chengfei Liu, Xiaochun Yang, Bin Wang, Jianxin Li, and Rui Zhou. 2016. Efficient Batch Processing for Multiple Keyword Queries on Graph Data. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24–28, 2016*, Snehasis Mukhopadhyay, ChengXiang Zhai, Elisa Bertino, Fabio Crestani, Javed Mostafa, Jie Tang, Luo Si, Xiaofang Zhou, Yi Chang, Yunyao Li, and Parikshit Sondhi (Eds.). ACM, 1261–1270. <https://doi.org/10.1145/2983323.2983806>
- [4] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 135–144.
- [5] Amin Milani Fard, Ebrahim Bagheri, and Ke Wang. 2019. Relationship Prediction in Dynamic Heterogeneous Information Networks. In *Advances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 11437)*, Leif Azzopardi, Benno Stein, Norbert Fuhr, Philipp Mayr, Claudia Hauff, and Djoerd Hiemstra (Eds.). Springer, 19–34. [https://doi.org/10.1007/978-3-030-15712-8\\_2](https://doi.org/10.1007/978-3-030-15712-8_2)
- [6] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The Web Conference 2020*. 2331–2341.
- [7] Yoav Goldberg and Omer Levy. 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722* (2014).
- [8] A Jordan et al. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems* 14, 2002 (2002), 841.
- [9] Mehdi Kargar, Lukasz Golab, Divesh Srivastava, Jaroslaw Szlichta, and Morteza Zihayat. 2020. Effective Keyword Search over Weighted Graphs. *IEEE Transactions on Knowledge and Data Engineering* early access (2020). <https://doi.org/10.1109/TKDE.2020.2985376>
- [10] Richard M. Karp. 1972. Reducibility Among Combinatorial Problems. In *Proceedings of a symposium on the Complexity of Computer Computations*. Plenum Press, New York, 85–103.
- [11] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [12] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *KDD*.
- [13] Theodoros Lappas, Kun Liu, and Evimaria Terzi. 2009. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, John F. Elder IV, Françoise Fogelman-Soulié, Peter A. Flach, and Mohammed Javeed Zaki (Eds.). ACM, 467–476. <https://doi.org/10.1145/1557019.1557074>
- [14] Theodoros Lappas, Kun Liu, and Evimaria Terzi. 2009. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, John F. Elder IV, Françoise Fogelman-Soulié, Peter A. Flach, and Mohammed Javeed Zaki (Eds.). ACM, 467–476. <https://doi.org/10.1145/1557019.1557074>
- [15] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States*, Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger (Eds.). 3111–3119. <https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html>
- [16] Radin Hamidi Rad, Hossein Fani, Mehdi Kargar, Jaroslaw Szlichta, and Ebrahim Bagheri. 2020. Learning to Form Skill-based Teams of Experts. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19–23, 2020*. 2049–2052.
- [17] Anna Sapienza, Palash Goyal, and Emilio Ferrara. 2019. Deep Neural Networks for Optimal Team Composition. *Frontiers Big Data* 2 (2019), 14. <https://doi.org/10.3389/fdata.2019.00014>
- [18] Fabian Schlieski, Jörg Schlotterer, and Michael Granitzer. 2020. Influence of Random Walk Parametrization on Graph Embeddings. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 12036)*, Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins (Eds.). Springer, 58–65. [https://doi.org/10.1007/978-3-030-45442-5\\_8](https://doi.org/10.1007/978-3-030-45442-5_8)
- [19] Mauro Sozio and Aristides Gionis. 2010. The community-search problem and how to plan a successful cocktail party. In *Proceedings ACM SIGKDD*. 939–948.
- [20] Yizhou Sun and Jiawei Han. 2012. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery* 3, 2 (2012), 1–159.
- [21] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. 2011. PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. *Proc. VLDB Endow.* 4, 11 (2011), 992–1003. <http://www.vldb.org/pvldb/vol4/p992-sun.pdf>
- [22] Yizhou Sun, Brandon Norick, Jiawei Han, Xifeng Yan, Philip S. Yu, and Xiao Yu. 2013. PathSelClus: Integrating Meta-Path Selection with User-Guided Object Clustering in Heterogeneous Information Networks. *ACM Trans. Knowl. Discov. Data* 7, 3 (2013), 11:1–11:23. <https://dl.acm.org/citation.cfm?id=2500492>
- [23] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *WSDM*. 495–503.
- [24] Morteza Zihayat, Aijun An, Lukasz Golab, Mehdi Kargar, and Jaroslaw Szlichta. 2017. Authority-based Team Discovery in Social Networks. In *Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21–24, 2017*, Volker Markl, Salvatore Orlando, Bernhard Mitschang, Periklis Andritsos, Kai-Uwe Sattler, and Sebastian Breß (Eds.). OpenProceedings.org, 498–501. <https://doi.org/10.5441/002/edbt.2017.54>