Translative Neural Team Recommendation

Abstract

Neural team recommendation has achieved state-of-the-art performance in forming teams of experts whose success in completing complex tasks is almost surely guaranteed. Specifically, the proposed models frame the problem as a Boolean multi-label classification, mapping a subset of required skills whose dense vector representations are transferred from expert collaboration graphs by graph neural networks to the sparse occurrence (multi-hot) vector representation of an optimum subset of experts using multilayer feedforward neural models. Such approaches, however, suffer from the curse of sparsity in the high-dimensional multi-hot vector representation of optimum experts in the output layer. In this paper, we propose to reformulate the team recommendation problem into a sequence prediction task and leverage transformers to map an input sequence of the required subset of skills onto an output sequence of the optimum subset of experts as the recommended team. Our experiments on four large-scale datasets from various domains, with distinct distributions of skills in teams, including `dblp` (computer science publications) and `uspt` (collection of issued patents) vs. `imdb` (movies) and `gith` (open-source software), show that our approach overall demonstrates consistently superior performance across key metrics such as `Precision`, `Recall`, `NDCG`, and `MAP`. Notably, our proposed method achieves substantial improvements in the top-2, top-5 and top-10 expert recommendations, with gains of up to 51x compared to the best-performing baseline in certain cases. This performance trend is consistent across all the datasets, highlighting the effectiveness and adaptability of our method in various domains.

1 Introduction

As modern projects have been surpassing the capacity of individuals, collaborative teams of experts have become vital in today's diverse landscape across academia, industry, law, freelancing, and healthcare, and the success of the project hinges on the effectiveness of the team. Assembling an effective team remains a complex challenge as it involves selecting individuals with the right expertise from usually a large pool of available candidates while ensuring their smooth interpersonal communication and collaboration. The manual approach has been shown to be tedious, error-prone, and suboptimal, as it is predisposed to hidden personal and societal biases [43], also falling short for an overwhelming number of experts, and failing to consider a multitude of criteria to optimize simultaneously [2]. Hence, A rich body of computational methods, from operations research [3, 17, 19, 29, 57, 64, 65, 69] social network analysis [22, 32, 55] and more recently, machine learning [9, 20, 48, 50] have been proposed to address team recommendation problem, also known as team allocation, team selection, team composition, and team configuration, among which neural models have brought state-of-the-art efficacy and efficiency due to the iterative and online learning procedure, and availability of training datasets. Team recommendation can also be seen as social information retrieval (Social IR), where the right group of experts, rather than relevant information, is desired to accomplish a task at hand.

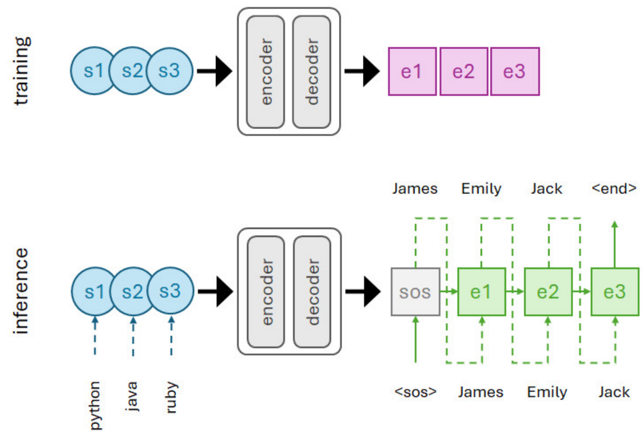

Figure 1: Overview of the sequence-2-sequence architecture.

By and large, neural models frame the team recommendation problem as a *multilabel* Boolean classification task, learning the distributions of experts and their skill sets in the context of successful teams in the past to draw future successful teams for experts who have a history of successful collaboration are statistically more likely to result in a successful team [51].

As seen in Figure **??**, in the input layer, neural models map a dense low-dimensional vector representation (embeddings) of a required subset of skills onto the output layer, which is an occurrence (multi-hot) vector representation of a successful (optimum) team. Hence, in the output layer, each expert is mapped to a label and would be recommended if their class's prediction probability is close to one. For instance, Rad et al. [48, 49] used embedding vectors of skills as the input for a variational Bayesian neural classifier. Such models, however, suffer from the curse of sparsity from the high-dimensional vector for the multi-hot vector representation of experts in the output layer. Due to the large number of labels (experts), neural classifiers, which learn their parameters based on average loss values over all labels, may underfit as the average loss becomes near zero over a very large number of experts. While researchers have tried weighted cross-entropy [50], and negative sampling heuristics [9] to address the sparsity in the output layer; such models still suffer from the lack of sufficient efficacy.

In this paper, we propose a sequence-to-sequence approach to the team recommendation problems. Unlike previous feedforward approaches, which first learn skill embeddings and then transfer them to a neural multi-label classifier, we reformulate the problem into an end-to-end sequence prediction task between pairs of (required subsets of skill → optimum subset of experts) and employ recurrent and/or transformer-based encoder-decoder neural models. As seen in Figure 1, given a successful team, we map it into a pair of its (required subset of skills → expert members). We then apply a recurrent neural network or a transformer to learn *** during training. During inference, given a test team with its subset of required skills, we predict a sequence of experts as the optimum recommended team. Our approach addresses the curse of sparsity and ***. Despite the similarity to sequence modelling, training recurrent neural models or transformers for team recommendation withholds challenges: (1) training datasets have their own unique skill and expert sets, and therefore, the pretraining/finetuning approach, which is successful for tasks like language modelling or translation, has limited applications for team recommendations. Moreover, a team recommender works with two distinct sets of skills and experts where their distributions over teams are highly domain-dependent. For instance, a majority of researchers may not author or co-author in fields other than their specialization, which is natural, but there will be some researchers who at least co-author in other fields. Unlike the imdb domain, this trend creates a highly skewed distribution of skills over teams. However, in imdb, the skill distribution is more uniform [20].

Transformers have gained significant traction for their effectiveness in underlying natural language processing tasks like language translation and beyond in cross-modal text-to-image generation or image-to-text captioning tasks and multi-modal video-to-text or vice versa tasks such as video summarization or video description tasks. Recently, they have found applications in recommendation systems like Transformer4Rec, a flexible and efficient library for state-of-the-art sequential and session-based recommendation [12]. However, to the best of our knowledge, no one has employed transformers for the team recommendation task before our study.

Our experiments across four large-scale datasets with varied distributions of teams over skills demonstrated that (1) ** (2) ** (3) ***.

2 Related Works

The work directly related to this paper can be broadly categorized into (1) neural sequence modelling and (2) neural team recommender systems.

2.1 Neural Sequence Modelling

While there has been a large body of work on sequence-to-sequence (seq-to-seq) models and transformers applied to a wide variety of tasks, such as natural language processing [13, 41, 52, 61], machine translation [1, 28, 61, 67], speech recognition [6, 8, 14, 24], and computer vision [4, 16, 42, 46], herein we focus on those in sequential recommendation systems [31, 58, 60, 68] as of particularly relevant to the team recommendation problem.

Hidasi et al. [26] was among the first who leveraged recurrent neural networks in recommender systems to address session-based recommendations where short/long sequences of user-item interactions were incorporated for better next-item recommendations, of which the de facto collaborative filtering methods or matrix factorization approaches fall short. They stack GRU units with a final feedforward layer whose loss was customized based on learning-to-rank losses, including pointwise and pairwise ranking losses and obtained marked improvements. In this line, Wu et al. [66] employed LSTM with softmax loss to capture the temporality of user-item interactions within time as sequences for better accuracy and speed.

Convolutional Sequence Embedding (Caser) is a CNN-based method for sequential recommendation tasks that models both short-term and long-term user preferences by embedding user-item interactions as sequences. It uses horizontal convolutions to capture temporal patterns across interactions and vertical convolutions to learn relationships between item features. This dual convolutional approach allows Caser to handle varying sequence lengths and produce compressed user embeddings, which are combined with item embeddings to predict the next item. Caser's structure is computationally efficient and scalable, offering strong performance in sparse data settings and as an alternative to RNNs for sequence modelling [31, 60].

A breakthrough for the sequential recommendation problem occurred when Kang et al. [31] proposed sas-rec, a transformer-based model with left-to-right self-attention layers, as in the Transformer [61] but with a point-wise ranking loss in its output feedforward layer. Later, Sun et al. [58] proposed bert4rec, where bert [5] with point-wise ranking loss has been trained on users' sequences of selected items using bidirectional self-attention layers.

Another powerful approach is Transformers4Rec [11, 12, 25, 31, 58], a library in NVIDIA's Merlin framework tailored for session-based and sequential recommendations using Transformer architectures. It extends the capabilities of the Hugging Face Transformers library to recommendation systems, offering flexible models that predict the next item in a sequence. By integrating with NVIDIA tools like NVTabular for preprocessing and Triton Inference Server for scalable deployment, it provides a robust, GPU-optimized solution for building fast and scalable recommendation models, particularly effective in next-item prediction tasks.

Despite extensive research on the application of sequence modelling in user-item recommendation, no work has addressed team recommendation using sequence modelling, and to the best of our knowledge, we are the first to bridge sequence learning strategies in the context of team recommendation. We presume that sequence modelling is a strong approach for team recommendations because it can handle varying lengths of input skills and output exerts. By using sequence-to-sequence pairs, we can retain the relationship between skills and experts, which is important for forming successful teams. This method has been shown to be more effective than traditional non-sequential techniques, such as variational and non-variational feedforward neural recommenders [20, 34, 48–51].

2.2 Neural Team Recommendation

Team recommendation has been a consistent research focus that draws interest from both computer science and social science audiences. Different approaches, both computational and noncomputational, have been employed in this area. For example, Paris et al. [45] studied the interactions and relationships among team members, while Stokols et al. [56] investigated effective teamwork dynamics. Despite their non-computational methods, their ultimate aim remains the same: to cultivate effective teams, often referred to as team selection, team formation, team configuration, or team allocation by various researchers. In contrast, the computational approaches are typically classified as follows: (1) search-based methods, where researchers identify successful teams through mathematical programming [3, 17, 19, 29, 57, 64, 65, 69] or expert network analysis [22, 32, 55]; (2) reinforcement-based methods [18, 39, 70], which treat team formation as a game, allowing experts to learn collaboration tactics through trial and error; and (3) learning-based methods [20, 34, 48–50, 54], which utilize neural networks to derive insights from previously successful teams. Although search-based and reinforcement-based techniques are theoretically sound, they often become inefficient in real-world scenarios involving numerous experts. Learning-based methods, however, have proven to be more efficient while maintaining effectiveness, largely due to their ability to learn and enhance their processes over time. This capability has established them as the preferred approach among researchers.

Within the learning-based paradigm, neural network approaches have evolved through three distinct generations. The first generation introduced basic non-variational feedforward networks [50], followed by more sophisticated variational Bayesian networks [10, 49, 50], and ultimately advanced to transfer learning methods using graph neural networks [34, 48, 54]. The evolution began with Rad et al. [50], who framed team recommendation as a multilabel classification task, implementing a simple feedforward network with one hidden layer that mapped input skills to output experts using standard cross-entropy loss. They later enhanced this approach by developing a variational Bayesian network [49, 50] that addressed popularity bias through incorporating Gaussian uncertainty into the model weights. Dashti et al. [9] further advanced this work by introducing negative sampling techniques, recognizing that experts lacking collaborative experience in specific skill areas would likely form unsuccessful teams. Given that popular experts dominated the training data, their method strategically selected groups of popular experts as negative examples to improve the model's discrimination ability. While these approaches showed promise, they shared a fundamental limitation: the assumption that experts could be selected independently, overlooking crucial team dynamics and social connections. Addressing this shortcoming, researchers turned to graph-based methods to capture collaborative relationships. Rad et al. [48] achieved state-of-the-art results by incorporating expert collaboration graphs into their model, using Dong et al.'s `metapath2vec` [15] to create skill embeddings from a heterogeneous graph connecting teams, experts, skills, and locations. Building on this foundation, Kaw et al. [34] leveraged deep graph infomax [62] to enhance skill embedding quality, employing a graph convolution network with an attention layer that achieved more effective skill representations in fewer training epochs.

Despite these advances, current neural methods still face significant challenges. The prevalent approach of treating team recommendation as a multilabel classification problem, where experts are chosen independently, fails to reflect real-world team dynamics. Additionally, both feedforward and Bayesian networks suffer from the use of high-dimensional multi-hot vector representations of experts in their output layers, leading to computational inefficiencies when dealing with large expert pools (numbering in the hundreds of thousands or millions). To address these limitations, we propose exploring a sequence-to-sequence approach, which transforms a dynamic-length input sequence of required skills into a dynamic-length output sequence of predicted experts.

3 Problem Definition

Given a set of skills $S = \{s_i\}$ and a set of experts $E = \{e_j\}$, a team is a tuple (s, e) where a *ranked* (e.g., in order of competence) subset of experts $e \subseteq E$ collectively covers a *ranked* (e.g., in order of importance) subset of required skills $s \subseteq S$ to accomplish a task at hand. Further, $T = \{(s, e)_k\}$ indexes all instances of successful teams. For a given set of skills s, the team recommendation problem aims at identifying an optimal subset of experts e such that their collaboration in the predicted team is successful. More concretely, the team recommendation problem is to learn a mapping function f of parameters θ from the powerset of skills to the powerset of experts such that $\forall(s, e) \in T; f_\theta(s) = e$.

4 Proposed Method

We propose to transform the team recommendation task into a machine translation task and estimate $f_\theta(s)$ on a *parallel dataset* whose pairs of sentences from a source language to a target language are pairs of $(s, e) \in T$, translating the ranked subset of skills $s = [s_1, .., s_i, .., s_n]$ as a source sentence to $e = [e_1, .., e_j, .., e_m]$ as a target sentence. We then employ a seq-to-seq encoder-decoder neural architecture to maximize the conditional probability $p(e|s)$ to learn f_θ [7, 27, 30, 44, 59]. The encoder maps the *sequence* of skills $[s_1, .., s_i, .., s_n]$ onto h_{i-1} and the decoder generates the sequence of experts $[e_1, .., e_j, .., e_m]$ from the h_{i-1}, one expert at a time, decomposing the conditional probability $p(e|s)$ as $\prod_{j=1} p(e_j|e_{<j}, s)$ and seeking the maximum probability among subsets of experts as an optimum team for s, i.e., $f_\theta(s) = e$. The probability of generating an expert at the decoder can be conditioned not only on h_{i-1} but also on all $h_{\leq i-1}$ at the encoder to enable the decoder to *attend* overall skills in the input sequence selectively (*global attention*) [44]. To reduce the computational complexity at the encoder and the decoder, a seq-2-seq model may have *no* recurrent connections to enable parallel calculation of $h_{\leq i}$ at the encoder and $h_{\leq i}$ at the decoder, like in transformers [61], which yielded promising performance on machine language translation and led to a large body of research on transformer-based language modeling [13, 53, 63].

5 Experiment Setup

This section details our experimental setup to answer the following research questions:

RQ1. Do the sequence-to-sequence neural architectures yield performance improvements over the state-of-the-art in the team recommendation task? In response, we benchmark translation-based team recommendation models against existing as well as state-of-the-art models [49, 50, 66].

RQ2. Which translation-based model performs the best (worst) in our proposed translative approach for the team recommendation task? In response, we compare a variety of sequence-based architectures, including vanilla recurrent models (rnn, *, *), convolutional seq-2seq (convs2s) [21], recurrent models with attention (rnn+attn) [71], and transformers [61] [*].

RQ3. How well does the translation-based model generalize across different domains in team recommendation? In response, we performed our experiment on four well-known benchmark datasets in team recommendation literature, including dblp, consisting of computer science publications and uspt collection of issued patents following long-tailed distribution of skills over teams, on the one hand, and imdb consisting of movies and github collection of open-source software repos, where skills were distributed uniformly among teams, on the other hand.

5.1 Datasets

Our testbed includes 4 benchmark datasets in team recommendation literature: dblp [38, 40], uspt [35], imdb [36], and gith [32, 33]. Each dataset was preprocessed to ensure a team consisted of more than one expert. The results of the dataset statistics are illustrated in Table 2.

→ In the **dblp** dataset, an expert is represented by an author, while a skill corresponds to the fields of study associated with a publication. A team, in this context, is mapped to a publication.

→ In the **uspt** dataset, an expert is represented by an inventor listed on a patent, while a skill corresponds to one of the subcategories in which the patent is classified. A team is mapped to a specific patent.

→ In the **imdb** dataset, an expert is represented by a member of the cast or crew, and a skill is mapped to the genre and subgenres of a movie. A team, therefore, corresponds to a movie.

→ In the **gith** dataset, an expert is represented by a contributor to a repository. A skill corresponds to one of the programming languages or software technologies, and a team is mapped to a repository on *github*.

For example, in dblp, a publication like "Attention is All You Need" (2017) represents a team where authors are the experts (e.g., Ashish Vaswani, Noam Shazeer, Niki Parmar, etc) and fields of study like "machine learning" and "natural language processing" are the skills. In uspt, a 2019 patent for "Methods and systems for using artificial intelligence to evaluate, correct, and monitor user attentiveness" (US11249544) maps to a team where inventors (e.g., Roberto Sicconi, Malgorzata Stys, and Tim Chinenov) are experts and patent subcategories like "G06F 9/54 Interprogram communication" and "G06T 7/11 Region-based segmentation" are skills. For imdb, the 2023 film "Oppenheimer" forms a team where cast and crew members (e.g., Christopher Nolan as director, Cillian Murphy as actor) are experts while genres/subgenres (e.g., "docudrama", "biography") represent skills. In gith, a popular 2023 repository "Next.js 14" represents a team where contributors (e.g., maintainers, key developers) are experts and technologies (e.g., "React", "TypeScript", "Node.js") serve as skills, with successful release and community adoption indicating team effectiveness.

5.2 Baseline

Our testbed includes baselines in two categories: (1) Neural team recommendation methods `bnn` [50] and `bnn_emb` [49]. The `bnn` and `bnn_emb` models feature a hidden layer size of 128 and utilize the ReLU activation function for their hidden layers and the sigmoid activation function for the output layers. We utilize `OpenTF` [10] for `bnn` and `bnn_emb`. (2) Translative models including vanilla recurrent neural network with attention (`rnn+attn`) [71], recurrent recommender network (`rrn`) [20, 66], convolutional seq-2-seq (`convs2s`) [21], and the Transformer [61]. Table **??** summarizes the hyperparameters and running settings used for our baselines. The code for `rrn` has kindly been provided by its authors. For the rest of the translative models, we used `OpenNMT-py` [37].

The baseline models (bnn, bnn_emb, and rnn) from [20] used a batch size of 128, learning rate of 0.1, and 20 epochs with Adam optimizer, employing a hidden layer size of 128 with ReLU and sigmoid activations. Our proposed models (convs2s, rnn+attn, transformer) used varying configurations, including a batch size of 8 for uspt dataset (128 otherwise), and learning rates following $d_{\text{model}}^{-0.5} \cdot step\_num \cdot warmup\_steps^{-1.5}$[61]. Full hyperparameter settings for each model can be found in their respective output folders–under each fold folder.

Table 1: Mapping of dataset attributes to team, expert, skill labels, and success.

dataset (success)	team(s,e) T	expert E	skill S
`dblp` (published)	publication	author	field of study
`uspt` (issued)	patent	inventor	subclass
`imdb` (produced)	movie	cast or crew	genre and subgenres
`gith` (released)	repository	contributor	programming languages and technologies

Table 2: Statistics of the raw and preprocessed datasets.

	`dblp` (1979-2018)		`uspt` (1976-2019)		`imdb` (1914-2020)		`gith` (2008-2022)	
	raw	prep.	raw	prep.	raw	prep.	raw	prep.
#teams	4.9M	99K	7.1M	152K	507K	32K	133K	46K
#experts	5.0M	14K	3.5M	13K	877K	2.0K	453K	1.2K
#skills	90K	30K	242K	67K	28	23	20	20
avg. #E/T	3.06	3.29	2.51	3.79	1.88	3.98	5.52	7.53
avg. #S/T	8.57	9.71	6.29	9.97	1.54	1.76	1.37	1.57

T: team, E: expert, S: skill, prep.: preprocessed

5.3 Evaluation Methodology and Metric

We randomly select 15% of teams for the test set and perform 3-fold cross-validation on the remaining teams for model training over 1 epoch, which results in one trained model per fold. Given a team (s, e) from the test set, we compare the sequence of experts e', predicted by the model of each fold, with the observed subset of experts e and report the average performance of models on all folds in terms of classification metrics including precision, recall, as well as information retrieval metrics including normalized discounted cumulative gain (ndcg), and mean average precision (map) at top-$\{2, 5, 10\}$. To compute these metrics, we use pytrec_eval [23] and scikit-learn [47]. The final results are obtained by averaging the

Table 3: Comparative results of feedforward vs. translative baselines. Bold and underlined values show best and second best, respectively.

dblp	Seq-2-Seq			Non Seq-2-Seq		
	t-rec	c-rec	r-rec	rrn	bnn	bnn_emb
%precision @2	**10.412**	2.500	3.618	0.057	0.112	0.057
%precision @5	**7.011**	1.612	2.358	0.066	0.129	0.039
%precision @10	**3.539**	0.824	1.199	0.071	0.125	0.047
%recall @2	**6.346**	1.507	2.170	0.035	0.067	0.038
%recall @5	**10.548**	2.418	3.511	0.099	0.191	0.063
%recall @10	**10.640**	2.476	3.575	0.212	0.370	0.155
%ndcg @2	**10.361**	2.477	3.582	0.054	0.108	0.048
%ndcg @5	**10.460**	2.428	3.518	0.081	0.156	0.052
%ndcg @10	**10.482**	2.449	3.539	0.133	0.240	0.096
%map @2	**5.946**	1.355	1.941	0.024	0.047	0.022
%map @5	**9.291**	2.001	2.879	0.041	0.079	0.028
%map @10	**9.321**	2.013	2.893	0.056	0.103	0.045

t-rec: transformer, c-rec: convs2s, r-rec: rnn+attn

uspt	Seq-2-Seq			Non Seq-2-Seq		
	t-rec	c-rec	r-rec	rrn	bnn	bnn_emb
%precision @2	**41.729**	28.572	23.973	0.066	0.366	0.024
%precision @5	**31.068**	24.653	17.787	0.077	0.412	0.038
%precision @10	**16.517**	15.238	9.472	0.091	0.375	0.065
%recall @2	**23.104**	13.910	12.987	0.035	0.161	0.014
%recall @5	**41.164**	28.817	23.036	0.098	0.451	0.050
%recall @10	**42.609**	33.760	23.890	0.221	0.814	0.137
%ndcg @2	**41.609**	28.361	23.915	0.066	0.365	0.022
%ndcg @5	**42.031**	30.032	23.823	0.088	0.453	0.041
%ndcg @10	**42.143**	31.414	23.827	0.148	0.609	0.087
%map @2	**22.405**	13.030	12.478	0.027	0.121	0.010
%map @5	**38.627**	24.460	21.357	0.043	0.203	0.019
%map @10	**39.759**	27.199	22.011	0.059	0.258	0.034

imdb	Seq-2-Seq			Non Seq-2-Seq		
	t-rec	c-rec	r-rec	rrn	bnn	bnn_emb
%precision @2	1.545	1.610	**1.699**	0.213	0.426	0.000
%precision @5	1.457	1.455	**1.480**	0.511	0.511	0.851
%precision @10	**0.904**	0.900	0.903	0.426	0.638	0.851
%recall @2	0.767	0.795	**0.819**	0.142	0.284	0.000
%recall @5	**1.809**	1.801	1.804	0.851	0.851	1.418
%recall @10	2.209	2.193	2.179	1.305	1.957	**2.837**
%ndcg @2	1.548	1.617	**1.700**	0.165	0.329	0.000
%ndcg @5	1.736	1.760	**1.788**	0.570	0.592	0.816
%ndcg @10	1.904	1.922	**1.933**	0.785	1.136	1.461
%map @2	0.617	0.651	**0.665**	0.071	0.142	0.000
%map @5	1.033	**1.049**	1.045	0.260	0.281	0.319
%map @10	1.091	**1.104**	1.098	0.315	0.439	0.627

gith	Seq-2-Seq			Non Seq-2-Seq		
	t-rec	c-rec	r-rec	rrn	bnn	bnn_emb
%precision @2	**32.160**	25.059	29.701	3.069	7.327	0.000
%precision @5	**21.605**	16.951	20.181	2.852	4.713	0.198
%precision @10	**12.710**	9.950	12.003	2.693	3.386	0.099
%recall @2	**13.854**	11.079	12.810	1.216	3.544	0.000
%recall @5	**22.291**	18.073	20.796	2.885	5.158	0.062
%recall @10	**24.087**	19.484	22.619	5.117	6.189	0.062
%ndcg @2	**32.429**	25.357	29.765	3.137	6.475	0.000
%ndcg @5	**28.254**	22.366	26.197	3.289	5.842	0.168
%ndcg @10	**26.990**	21.585	25.126	4.234	6.267	0.109
%map @2	**12.955**	10.516	11.976	1.010	2.342	0.000
%map @5	**19.321**	15.562	17.998	1.571	3.082	0.021
%map @10	**20.798**	16.643	19.484	2.163	3.384	0.021

performance metrics across all three folds for a robust evaluation of the model's predictive capabilities while minimizing fold-specific variation in the data distribution.

The transformer model predicts experts' order through its parallel self-attention mechanism, where each selection considers both the full skill sequence and all previously predicted experts simultaneously via the attention weights. One of the advantages of this is that the model can capture dependencies regardless of their sequence position. The convs2s model, on the other hand, processes skill-expert relationships through stacked convolutional layers that create hierarchical representations. Lower layers capture local skill-expert matches, while deeper layers learn broader team composition patterns by building on this information. This hierarchical structure allows the model to efficiently capture relationships between distant elements in the sequence. The rnn with attention learns the expert selection sequentially, and in the hidden state, the expert choice is updated using attention to focus on which expert is relevant to which skill. The prediction in rnn depends on both the cumulative history in the hidden state and attended skill information. After each model processes the importance of experts to certain skills in their own methods, the order of the predicted experts is influenced by the order of the required input skill.

5.4 Results

Foremost, we acknowledge that baselines achieve low values of evaluation metrics for practical applications of team recommendation, which is primarily due to the simplicity of the neural model architectures and the small number of training epochs given the intensive computational demands for such methods coupled with our limited computational resources; metric values are reported in % for ease of readability and comparison. Our main goal is to showcase the optimum solution setting, seq-2-seq vs. multilabel classification, for team recommendation.

In response to **RQ1**, i.e., whether sequence-to-sequence neural architectures yield better performance versus feedforward neural architectures for team recommendation, from Table 4, *all* the sequence-to-sequence models statistically significantly outperform the feedforward models across metrics in different datasets, which can be attributed to recommending the next expert based (conditioned) on the previously recommended experts in the output sequence, allowing for more robust prediction despite the inherent challenges posed by sparse activations in the output layer. Notably, translative models' relative improvements have been within the impressive range of 10x up to 100x, suggesting the right track for the most suitable neural architecture for the team recommendation task. The only exception is the recurrent recommender network (rrn), which performs on par or even poorer than the feedforward baselines and will be discussed more in **RQ2**.

In response to **RQ2**, that is looking into the translative models for the best (worst) sequence-to-sequence architectures, from Table

4, we observe that the transformer consistently and statistically significantly outperforms other translative models across *all* datasets and metrics. This is because the Transformer model benefits from its three key architecture designs: (1) its self-attention mechanism allows to capture the relevance of experts to skills independently and simultaneously, (2) the encoder-decoder structure effectively captures input context and output context regardless of position in the sequence due to (1), and (3) the parallel processing helps in handling with varying team sizes and skill distribution. The result is a consistent effectiveness between uniformly distributed and long-tailed datasets as observed in the result tables. The second-best translative model, however, depends on the underlying distribution of skills over teams (the input sequence) in a dataset. In dblp and uspt, where the distribution of skills is long-tailed, convs2s is the runner up whereas in imdb and gith, where the skills are uniformly distributed in teams, vanilla recurrent neural network with attention (rnn+attn) is the second-best. The convs2s model processes the input tokens simultaneously and uses its multi-hop attention mechanism, combined with its hierarchical convolutional structure, to capture relevant expert-to-skill information which is more effective than rnn with the attention mechanism that learns sequentially in datasets that are long-tailed, as observed. However, in uniformly distributed datasets, the more complex architecture of the convs2s model does not provide an advantage when the frequency of skills appearing is similar in the dataset. Lastly, the worst translative model is the recurrent recommender network (rrn), and as mentioned in RQ1, falls short of consistent outperformance even compared to feedforward baselines across datasets and metrics. Specifically, rrn performs the poorest in dblp, uspt, and gith, and only in imdb, it shows performance improvement over feedforward baselines because rrn uses an LSTM autoregressive model, but without attention mechanism, which means all relevant information must be in a fixed-size hidden state to be used as historical information or else it'd not be considered. Moreover, the other three datasets besides imdb are long-tailed, where few skills appear frequently and while most are rare, which makes the LSTM model struggle to keep these within the fixed hidden states for use as historical context—thus contributing to the poor performance.

To answer **RQ3**, i.e., whether the outperformance of translative models can generalize to various domains, from Table 3, the results demonstrate superior performance compared to the baselines across all four datasets across metrics. However, the *extent* of improvement depends on the underlying dataset, esp. the distribution of skills over teams. In imdb and gith, where the skills are distributed uniformly in the input sequence, the improvement extends to about 100x on the average across metrics. In contrast, when the skills are distributed in a long-tailed fashion, the improvement increases to about 24x. In both types of skills distribution, the improvement is at least 2x. This showcases how well the translative models generalize to various domains over the baselines. The results show the different strengths of the translative models' different architecture approaches based on the type of dataset trained on, and due to this, it makes them all relevant to the research in neural machine translation for team formation tasks.

6 Conclusion and Future Work

In this paper, we proposed to reformulate the team recommendation problem as a sequence prediction task and to use sequence-2-sequence neural architectures, as opposed to the existing (variational) feedforward architectures. Our findings showed that sequence-to-sequence neural architectures yield performance improvements and are better architectures. Among sequence-to-sequence models, while the transformer variation has been the best model across datasets and metrics, the runner-up model depends on the distribution of skills over teams. Specifically, in datasets where the distributions of skills are *long-tailed*, convs2s was the runner-up whereas in datasets where the skills are *uniformly* distributed in teams, vanilla recurrent neural network with attention (rnn+attn) was the second-best. Our future work includes investigating sequence-2-sequence models in the presence of additional contextual factors, such as geolocation (geo-aware team recommendation).

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1409.0473

[2] Rodrigo Borges, Otto Sahlgren, Sami Koivunen, Kostas Stefanidis, Thomas Olsson, and Arto Laitinen. 2023. Multi-Objective Fairness in Team Assembly. In *New Trends in Database and Information Systems - ADBIS 2023 Short Papers, Doctoral Consortium and Workshops: AIDMA, DOING, K-Gals, MADEISD, PeRS, Barcelona, Spain, September 4-7, 2023, Proceedings (Communications in Computer and Information Science, Vol. 1850)*, Alberto Abelló, Panos Vassiliadis, Oscar Romero, Robert Wrembel, Francesca Bugiotti, Johann Gamper, Genoveva Vargas-Solar, and Ester Zumpano (Eds.). Springer, 106–116. https://doi.org/10.1007/978-3-031-42941-5_10

[3] Manoel B. Campêlo, Tatiane Fernandes Figueiredo, and Ana Silva. 2020. The sociotechnical teams formation problem: a mathematical optimization approach. *Ann. Oper. Res.* 286, 1 (2020), 201–216. https://doi.org/10.1007/S10479-018-2759-5

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-End Object Detection with Transformers. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12346)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 213–229. https://doi.org/10.1007/978-3-030-58452-8_13

[5] Xiaoyang Chen, Kai Hui, Ben He, Xianpei Han, Le Sun, and Zheng Ye. 2021. Co-BERT: A Context-Aware BERT Retrieval Model Incorporating Local and Query-specific Context. *CoRR* abs/2104.08523 (2021). arXiv:2104.08523 https://arxiv.org/abs/2104.08523

[6] Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani. 2018. State-of-the-Art Speech Recognition with Sequence-to-Sequence Models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*. IEEE, 4774–4778. https://doi.org/10.1109/ICASSP.2018.8462105

[7] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1724–1734. https://doi.org/10.3115/V1/D14-1179

[8] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-Based Models for Speech Recognition. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (Eds.). 577–585. https://proceedings.neurips.cc/paper/2015/hash/1068c6e4c8051cfd4e9ea8072e3189e2-Abstract.html

[9] Arman Dashti, Saeed Samet, and Hossein Fani. 2022. Effective Neural Team Formation via Negative Samples. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, Mohammad Al Hasan and Li Xiong (Eds.). ACM, 3908–3912. https://doi.org/10.1145/3511808.3557590

[10] Arman Dashti, Karan Saxena, Dhwani Patel, and Hossein Fani. 2022. OpeNTF: A Benchmark Library for Neural Team Formation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, Mohammad Al Hasan and Li Xiong (Eds.). ACM, 3913–3917. https://doi.org/10.1145/3511808.3557526

[11] Gabriel de Souza Pereira Moreira, Sara Rabhi, Ronay Ak, Md Yasin Kabir, and Even Oldridge. 2021. Transformers with multi-modal features and post-fusion context for e-commerce session-based recommendation. *CoRR* abs/2107.05124 (2021). arXiv:2107.05124 https://arxiv.org/abs/2107.05124

[12] Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeongmin Lee, Ronay Ak, and Even Oldridge. 2021. Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation. In *RecSys '21: Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September 2021 - 1 October 2021*, Humberto Jesús Corona Pampín, Martha A. Larson, Martijn C. Willemsen, Joseph A. Konstan, Julian J. McAuley, Jean Garcia-Gathright, Bouke Huurnink, and Even Oldridge (Eds.). ACM, 143–153. https://doi.org/10.1145/3460231.3474255

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/V1/N19-1423

[14] Linhao Dong, Shuang Xu, and Bo Xu. 2018. Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*. IEEE, 5884–5888. https://doi.org/10.1109/ICASSP.2018.8462506

[15] 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, Halifax, NS, Canada, August 13 - 17, 2017*. ACM, 135–144. https://doi.org/10.1145/3097983.3098036

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=YicbFdNTTy

[17] Edmund H. Durfee, James C. Boerkoel Jr., and Jason Sleight. 2014. Using hybrid scheduling for the semi-autonomous formation of expert teams. *Future Gener. Comput. Syst.* 31 (2014), 200–212. https://doi.org/10.1016/J.FUTURE.2013.04.008

[18] Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob N. Foerster, and Shimon Whiteson. 2023. SMACv2: An Improved Benchmark for Cooperative Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.). http://papers.nips.cc/paper_files/paper/2023/hash/764c18ad230f9e7bf6a77ffc2312c55e-Abstract-Datasets_and_Benchmarks.html

[19] José G. M. Esgario, Iago E. da Silva, and Renato A. Krohling. 2019. Application of Genetic Algorithms to the Multiple Team Formation Problem. *CoRR* abs/1903.03523 (2019). arXiv:1903.03523 http://arxiv.org/abs/1903.03523

[20] Hossein Fani, Reza Barzegar, Arman Dashti, and Mahdis Saeedi. 2024. A Streaming Approach to Neural Team Formation Training. In *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 14608)*, Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (Eds.). Springer, 325–340. https://doi.org/10.1007/978-3-031-56027-9_20

[21] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017 (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 1243–1252. http://proceedings.mlr.press/v70/gehring17a.html

[22] Kiarash Golzadeh, Lukasz Golab, and Jaroslaw Szlichta. 2024. Explaining Expert Search and Team Formation Systems with ExES. *CoRR* abs/2405.12881 (2024). https://doi.org/10.48550/ARXIV.2405.12881 arXiv:2405.12881

[23] Christophe Van Gysel and Maarten de Rijke. 2018. Pytrec_eval: An Extremely Fast Python Interface to trec_eval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (Eds.). ACM, 873–876. https://doi.org/10.1145/3209978.3210065

[24] Awni Y. Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. 2014. Deep Speech: Scaling up end-to-end speech recognition. *CoRR* abs/1412.5567 (2014). arXiv:1412.5567 http://arxiv.org/abs/1412.5567

[25] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (Eds.). ACM, 843–852. https://doi.org/10.1145/3269206.3271761

[26] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1511.06939

[27] Sébastien Jean, KyungHyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On Using Very Large Target Vocabulary for Neural Machine Translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. The Association for Computer Linguistics, 1–10. https://doi.org/10.3115/V1/P15-1001

[28] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *CoRR* abs/1611.04558 (2016). arXiv:1611.04558 http://arxiv.org/abs/1611.04558

[29] Sunny Joseph Kalayathankal, John T Abraham, and Joseph Varghese Kureethara. 2019. A Fuzzy Approach To Project Team Selection. *International Journal of Scientific Technology Research* 8 (2019).

[30] Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 1700–1709. https://aclanthology.org/D13-1176/

[31] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*. IEEE Computer Society, 197–206. https://doi.org/10.1109/ICDM.2018.00035

[32] Mehdi Kargar and Aijun An. 2011. Discovering top-k teams of experts with/without a leader in social networks. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, Craig Macdonald, Iadh Ounis, and Ian Ruthven (Eds.). ACM, 985–994. https://doi.org/10.1145/2063576.2063718

[33] Mehdi Kargar, Lukasz Golab, Divesh Srivastava, Jaroslaw Szlichta, and Morteza Zihayat. 2022. Effective Keyword Search Over Weighted Graphs. *IEEE Trans. Knowl. Data Eng.* 34, 2 (2022), 601–616. https://doi.org/10.1109/TKDE.2020.2985376

[34] Sagar Kaw, Ziad Kobti, and Kalyani Selvarajah. 2023. Transfer Learning with Graph Attention Networks for Team Recommendation. In *International Joint Conference on Neural Networks, IJCNN 2023, Gold Coast, Australia, June 18-23, 2023*. IEEE, 1–8. https://doi.org/10.1109/IJCNN54540.2023.10191717

[35] Peter Keane, Faisal Ghaffar, and David Malone. 2020. Using machine learning to predict links and improve Steiner tree solutions to team formation problems - a cross company study. *Appl. Netw. Sci.* 5, 1 (2020), 57. https://doi.org/10.1007/S41109-020-00306-X

[36] Abeer Khan, Lukasz Golab, Mehdi Kargar, Jaroslaw Szlichta, and Morteza Zihayat. 2020. Compact group discovery in attributed graphs and social networks. *Inf. Process. Manag.* 57, 2 (2020), 102054. https://doi.org/10.1016/J.IPM.2019.102054

[37] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of ACL 2017, System Demonstrations*. Association for Computational Linguistics, Vancouver, Canada, 67–72. https://www.aclweb.org/anthology/P17-4012

[38] Yue Kou, Derong Shen, Quinn Snell, Dong Li, Tiezheng Nie, Ge Yu, and Shuai Ma. 2020. Efficient Team Formation in Social Networks based on Constrained Pattern Graph. In *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020*. IEEE, 889–900. https://doi.org/10.1109/ICDE48307.2020.00082

[39] Karol Kurach, Anton Raichuk, Piotr Stanczyk, Michal Zajac, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. 2020. Google Research Football: A Novel Reinforcement Learning Environment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 4501–4510. https://doi.org/10.1609/AAAI.V34I04.5878

[40] 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

[41] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[42] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 9992–10002. https://doi.org/10.1109/ICCV48922.2021.00986

[43] Hamed Loghmani and Hossein Fani. 2023. Bootless Application of Greedy Re-ranking Algorithms in Fair Neural Team Formation. In *Advances in Bias and Fairness in Information Retrieval - 4th International Workshop, BIAS 2023, Dublin, Ireland, April 2, 2023, Revised Selected Papers (Communications in Computer and Information Science, Vol. 1840)*, Ludovico Boratto, Stefano Faralli, Mirko Marras, and Giovanni Stilo (Eds.). Springer, 108–118. https://doi.org/10.1007/978-3-031-37249-0_9

[44] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton (Eds.). The Association for Computational Linguistics, 1412–1421. https://doi.org/10.18653/V1/D15-1166

[45] Carol R. Paris, Eduardo Salas, and Janis A. Cannon-Bowers. 2000. Teamwork in multi-person systems: a review and analysis. *Ergonomics* 43 (2000), 1052 – 1075. https://api.semanticscholar.org/CorpusID:41152229

[46] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image Transformer. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 4052–4061. http://proceedings.mlr.press/v80/parmar18a.html

[47] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12 (2011), 2825–2830. https://doi.org/10.5555/1953048.2078195

[48] Radin Hamidi Rad, Ebrahim Bagheri, Mehdi Kargar, Divesh Srivastava, and Jaroslaw Szlichta. 2021. Retrieving Skill-Based Teams from Collaboration Networks. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 2015–2019. https://doi.org/10.1145/3404835.3463105

[49] Radin Hamidi Rad, Hossein Fani, Ebrahim Bagheri, Mehdi Kargar, Divesh Srivastava, and Jaroslaw Szlichta. 2024. A Variational Neural Architecture for Skill-based Team Formation. *ACM Trans. Inf. Syst.* 42, 1 (2024), 7:1–7:28. https://doi.org/10.1145/3589762

[50] 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*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 2049–2052. https://doi.org/10.1145/3340531.3412140

[51] Radin Hamidi Rad, Shirin Seyedsalehi, Mehdi Kargar, Morteza Zihayat, and Ebrahim Bagheri. 2022. A Neural Approach to Forming Coherent Teams in Collaboration Networks. In *Proceedings of the 25th International Conference on Extending Database Technology, EDBT 2022, Edinburgh, UK, March 29 - April 1, 2022*, Julia Stoyanovich, Jens Teubner, Paolo Guagliardo, Milos Nikolic, Andreas Pieris, Jan Mühlig, Fatma Özcan, Sebastian Schelter, H. V. Jagadish, and Meihui Zhang (Eds.). OpenProceedings.org, 2:440–2:444. https://doi.org/10.48786/EDBT.2022.37

[52] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[53] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67. https://jmlr.org/papers/v21/20-074.html

[54] 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

[55] Mauro Sozio and Aristides Gionis. 2010. The community-search problem and how to plan a successful cocktail party. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*, Bharat Rao, Balaji Krishnapuram, Andrew Tomkins, and Qiang Yang (Eds.). ACM, 939–948. https://doi.org/10.1145/1835804.1835923

[56] Daniel Stokols, Kara Hall, B.K. Taylor, and R.P. Moser. 2008. The science of team science. *Am. J. Preventive Med.* 35 (01 2008), S78–S89.

[57] Damjan Strnad and Nikola Guid. 2010. A fuzzy-genetic decision support system for project team formation. *Appl. Soft Comput.* 10, 4 (2010), 1178–1187. https://doi.org/10.1016/J.ASOC.2009.08.032

[58] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, 1441–1450. https://doi.org/10.1145/3357384.3357895

[59] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). 3104–3112. https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html

[60] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, Yi Chang, Chengxiang Zhai, Yan Liu, and Yoelle Maarek (Eds.). ACM, 565–573. https://doi.org/10.1145/3159652.3159656

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[62] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep Graph Infomax. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=rklz9iAcKQ

[63] Suzan Verberne. 2023. Pretrained Transformers for Text Ranking: BERT and Beyond. *Comput. Linguistics* 49, 1 (2023), 253–255. https://doi.org/10.1162/COLI_R_00468

[64] Lin Wang, Yifeng Zeng, Bilian Chen, Yinghui Pan, and Langcai Cao. 2020. Team Recommendation Using Order-Based Fuzzy Integral and NSGA-II in StarCraft. *IEEE Access* 8 (2020), 59559–59570. https://doi.org/10.1109/ACCESS.2020.2982647

[65] Hyeongon Wi, Seungjin Oh, Jungtae Mun, and Mooyoung Jung. 2009. A team formation model based on knowledge and collaboration. *Expert Syst. Appl.* 36, 5 (2009), 9121–9134. https://doi.org/10.1016/J.ESWA.2008.12.031

[66] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. 2017. Recurrent Recommender Networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, Maarten de Rijke, Milad Shokouhi, Andrew Tomkins, and Min Zhang (Eds.). ACM, 495–503. https://doi.org/10.1145/3018661.3018689

[67] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR* abs/1609.08144 (2016). arXiv:1609.08144 http://arxiv.org/abs/1609.08144

[68] Fajie Yuan, Xiangnan He, Haochuan Jiang, Guibing Guo, Jian Xiong, Zhezhao Xu, and Yilin Xiong. 2020. Future Data Helps Training: Modeling Future Contexts for Session-based Recommendation. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen (Eds.). ACM / IW3C2, 303–313. https://doi.org/10.1145/3366423.3380116

[69] Armen Zakarian and Andrew Kusiak. 1999. Forming teams: An analytical approach. *IIE Transactions* 31 (01 1999), 85–97. https://doi.org/10.1023/A:1007580823003

[70] Yifan Zang, Jinmin He, Kai Li, Haobo Fu, Qiang Fu, Junliang Xing, and Jian Cheng. 2023. Automatic Grouping for Efficient Cooperative Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.). http://papers.nips.cc/paper_files/paper/2023/hash/

906c860f1b7515a8ffec02dcdac74048-Abstract-Conference.html
[71] Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. 2016. Deep Recurrent Models with Fast-Forward Connections for Neural Machine Translation. *Trans. Assoc. Comput. Linguistics* 4 (2016), 371–383. https://doi.org/10.1162/TACL_A_00105