Say the Task, Build the Team: Prompt-Based Team Formation

Lingling Zhang[1], Radin Hamidi Rad[2], Morteza Zihayat[1], and Ebrahim Bagheri[2]

[1] Toronto Metropolitan University, Toronto Ontario M5B 2K3, Canada, {zhll,mzihayat}@torontomu.ca,
[2] University of Toronto, Toronto Ontario M5S 3G6, Canada {radin.rad,ebrahim.bagheri}@utoronto.ca

Abstract. The problem of assembling effective expert teams based on project needs is central to expert networks such as LinkedIn. However, current team formation methods typically depend on keyword-matching techniques that fail to capture the nuanced semantics of natural language project descriptions. This results in inadequate modeling of required expertise and suboptimal team selection. Addressing this gap, we propose a contextual, prompt-driven framework for team formation that infers latent expertise from rich textual descriptions of project goals. Our approach fine-tunes a T5-Large sequence-to-sequence model to translate project prompts into expert team compositions by benefiting from enhanced expertise annotations. To facilitate this task, we curate, and publicly release, a dataset based on DBLP V14 collection, augmented with high-confidence expertise labels generated by large language models. Experimental results across multiple evaluation metrics show that our proposed model outperforms existing state-of-the-art baselines, underscoring the importance of contextualized representations in expert discovery and team assembly.

Keywords: Team Formation, Expert Network, Sequential Learning

1 Introduction

Effective team formation is a central challenge in collaborative environments spanning academia, industry, and online platforms. In the context of expert networks, the team formation problem refers to identifying a subset of professionals from a *social graph* such that their collective expertise satisfies the skill requirements of a given project [19]. Unlike the expert finding problem, which returns a ranked list of individuals most relevant to a given query, the team formation problem additionally requires that the selected experts not only collectively fulfill the required skills but also exhibit strong connectivity within the underlying collaboration network [12, 22, 23]. Since finding an ideal team is an NP-hard problem [8, 14, 19, 30, 39], many studies have proposed approximate algorithms [2, 4, 11, 15, 16, 18, 19], neural network-based approaches [6, 9, 10, 17, 24,

26, 27, 29, 30] and most recently, Large Language Model (LLM)-based architectures [5, 35] to identify optimal teams while balancing multiple objectives (e.g., budget, collaboration) efficiently.

Existing methods for team formation typically assume that the required skills are explicitly specified. However, this assumption breaks down in real-world scenarios, where project descriptions are often expressed in natural language without a clear enumeration of required skills [13]. In such cases, identifying suitable experts necessitates a deeper semantic understanding of the project's context beyond surface-level keyword matching. A few recent works have begun addressing this challenge by leveraging auxiliary metadata, such as project titles or brief textual summaries, to infer the underlying expertise needs [10, 24, 31].

In this light, we believe that prompt-based retrieval and generation frameworks [36, 38] offer a promising direction for team formation, as they allow models to process detailed natural language inputs more faithfully reflecting how project requirements are articulated in real-world contexts [3]. Unlike rigid, skill-list-based formulations, contextualized prompts encode nuanced semantic cues and often implicitly capture the expertise required for a given task. Compared to traditional keyword-based or short-text inputs, they provide a more expressive and complete representation of project intent. Despite these advantages, most existing approaches to the team formation problem remain limited to shallow representations, relying on explicit keyword sets [15, 18] or metadata fields such as titles and predefined research areas [9, 10, 31]. These formulations fall short when project descriptions are unstructured and required skills are rarely enumerated explicitly.

Another key bottleneck in advancing prompt-based team formation is the lack of suitable datasets. Existing resources rely on coarse proxies, such as co-authorship or paper-author links, that fail to capture the alignment between natural language project descriptions and the required team skills. As a result, supervised models in this space remain underdeveloped. The development of better methods would require datasets that explicitly link rich textual prompts to suitable expert teams while preserving the implicit semantics of required skills.

To overcome the limitations of existing team formation approaches, we introduce *Prompt-based Team Formation (PTF)*, where the goal is to generate an expert team from a natural language prompt (e.g., a project description) such that the selected members collectively fulfill the implicit skill requirements[3]. This task reflects how project needs are communicated in real-world settings and addresses the gap between textual project intent and structured expertise representations. Our key contributions are as follows:

1. We construct and publicly release a dataset based on the DBLP V14 collection [34], augmented with high-confidence expertise annotations derived from large language models. This resource supports both prompt-based and traditional team formation methods.

[3]Our dataset and code are publicly available at https://github.com/littlebeanbean7/Prompt-Based-Team-Formation

2. We formulate the task as a conditional sequence-to-sequence (seq-to-seq) generation problem, enabling models to map project prompts directly to coherent expert teams.
3. We conduct comprehensive empirical evaluations, demonstrating that our approach outperforms state-of-the-art baselines across multiple metrics, highlighting the value of contextualized representations in expert discovery.

2 Related Work

2.1 Team Formation and Expert Retrieval

The team formation problem has been studied across various domains, with early efforts focusing on optimization and graph-based methods. In recent years, machine learning approaches have gained traction, leveraging the power of neural networks to better model team dynamics and improve performance.

The team formation problem was initially addressed using optimization-based techniques. Early work in this area focused on mathematical models to optimize team compositions. Baykasoglu et al. [2] proposed a fuzzy optimization model to select teams by considering various factors like skills, compatibility, and availability. Fitzpatrick and Askin [11] proposed a heuristic solution method.

As the field advanced, researchers began to incorporate social networks and game-theoretic principles into team formation. Wi et al [37] introduced a hybrid approach that integrates genetic algorithms with social network analysis to form project teams. Lappas et al. [19] tackled the problem of forming a team that collectively covers a required set of skills while minimizing communication costs within a social network. Zihayat et al. [39] proposed to form expert teams by considering both the required skills and the authority levels of individuals within a social network.

With the rise of machine learning, a variety of neural networks were utilized to solve the team formation problem. Sapienza et al. [33] introduced a deep learning framework designed to recommend optimal teammates in cooperative environments. Rad et al. [29] presented a variational Bayesian neural network architecture to form expert teams that collectively cover a set of required skills, leveraging historical collaboration data. The model handles data sparsity and scales efficiently to large expert networks by incorporating uncertainty through variational inference. Rad et al. [26] proposed a heterogeneous graph utilizing embedding methods to capture the complex relationships in the collaboration network. Kaw et al. [17] applied graph attention networks (GATs) to team formation, demonstrating the ability of attention mechanisms to focus on the most relevant skills and expertise when forming teams. Fani et al. [9] introduced a temporal training strategy for neural models to capture the evolution of experts' skills and collaboration ties over time. Although neural approaches have advanced the team formation problem, they still treat expert selection as a set of independent decisions by framing it as a multilabel classification task. This simplification fails to capture the complex dynamics of real-world teams. Recently, LLM-based architectures have been applied to team formation problem. Dara et al. [5] integrated retrieval-augmented generation (RAG) techniques

with deep learning models. Thang et al. [35] attempted to introduce sequence to team formation. They found that Transformer-based seq-to-seq models are the most powerful. Despite the potential of LLMs to handle rich textual data, both of the recent studies still relied on keyword-based representations of project requirements. They did not process natural language project descriptions or reason over implicit expertise needs embedded in free-text prompts when forming teams. And the seq-to-seq model by Thang et al. [35] did not consider the order of skills.

2.2 Prompt-Based Generation and Sequence Models

Prompt-based learning has emerged as a powerful paradigm for leveraging pretrained language models with minimal task-specific supervision. Early approaches such as GPT-3 [3] demonstrated that large language models can perform various tasks through few-shot prompting without fine-tuning. This inspired a shift from model-centric to data-centric paradigms. Language models fine-tuned with prompts have shown effectiveness in complex retrieval, reasoning, and recommendation tasks [7, 20, 32]. In the context of team formation, this approach enables the model to better understand skill-expert matching by framing the input as a structured prompt. Our method follows this paradigm by fine-tuning a pretrained seq-to-seq model on prompt-structured inputs for the team recommendation task. Our work proposes a novel formulation of team formation as a conditional sequence generation task, where input prompts include both structured annotations (expertise) and contextual project descriptions, and the output is an ordered team of expert identifiers. This shift enables the model to reason jointly over explicit and latent signals, leveraging the expressive power of modern seq-to-seq models. Unlike prior graph- or keyword-based approaches, our method can operate directly on natural language inputs and generalize to more realistic, less constrained team formation scenarios.

3 PTF: A Prompt-based Team Formation Model

To address the task of *Contextual Prompt-Driven Team Formation*, we propose a generative framework that takes a natural language project description as input and synthesizes a team of experts whose aggregated expertise satisfies the implicit and explicit skill requirements embedded in the prompt. We formalize *Contextual Prompt-Driven Team Formation* as a conditional seq-to-seq generation problem. Given a project p_i described by natural language text d_i, and an associated set of inferred expertise annotations E_i, the objective is to generate a team $T_i = \{a_1, a_2, \ldots, a_k\}$ such that each expert a_j contributes at least one relevant skill and the team's aggregated expertise satisfies both the implicit and explicit requirements embedded in the prompt.

To this end, we develop a model that takes as input a contextualized representation of the project, comprising both free-text description and structured expertise cues—and outputs an ordered sequence of expert identifiers. We implement this using a fine-tuned T5 model [32], although the framework is model-agnostic

and compatible with alternative encoder-decoder or decoder-only language models. Formally, each input is encoded as:

$$\mathbf{X} = \mathcal{T}\left(\mathcal{C}\left(\texttt{``Queries: ''}, E_i, \texttt{`` Context: ''}, d_i\right)\right)$$

where $\mathcal{C}(\cdot)$ denotes the string-level concatenation of structured expertise and textual description using fixed input prompts, and $\mathcal{T}(\cdot)$ denotes tokenization using the T5 tokenizer. The inclusion of E_i serves as an inductive prior, offering a high-level abstraction of project requirements that complements the natural language signal in d_i, thereby enhancing the model's capacity to align tasks with appropriate expert candidates.

The output sequence corresponds to the predicted expert identifiers $\hat{T}_i = \{\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_{k_i}\}$, where each token $\hat{a}_j$ denotes an expert selected to form the team for project p_i. The model learns to maximize the likelihood of generating the correct sequence T_i, conditioned on the input pair (d_i, E_i), where d_i is the natural language project description and E_i is the set of structured expertise annotations inferred for the task. We formalize this as a sequence-level optimization problem using the standard cross-entropy loss:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \sum_{j=1}^{k_i} \log P(a_j \mid d_i, E_i; \theta)$$

Here, N is the number of training samples, $k_i = |T_i|$ denotes the length of the ground-truth team for project i, and $P(a_j \mid d_i, E_i; \theta)$ is the probability of generating the j-th expert in the sequence given the input prompt and current model parameters θ. This formulation captures the autoregressive nature of the generation task, enabling the model to condition each expert prediction on both the project context and previously generated experts.

The full training pipeline is detailed in Algorithm 1. During each iteration, project prompts and corresponding ground-truth teams are converted into token sequences via the modified tokenizer and passed through the encoder-decoder architecture using teacher forcing. Gradients are computed via backpropagation on the weighted sequence-level loss and used to update the model parameters via AdamW [21] optimization. The procedure is fully compatible with any autoregressive language model supporting conditional decoding, including both encoder-decoder and decoder-only variants.

Overall, the training setup is designed to jointly leverage semantic richness from natural language descriptions and structural signal from inferred expertise annotations. It ensures that the model not only learns to decode plausible expert identifiers but also internalizes the nuanced skill composition required for coherent and functionally aligned team formation.

4 Proposed Dataset Construction

We construct a dataset that includes detailed project descriptions, inferred expertise annotations, and relevant metadata to support expert team formation based on contextual prompts.

Algorithm 1 Prompt-Based Team Formation Training

Input: Training set $\mathcal{D} = \{(d_i, E_i, T_i)\}_{i=1}^{N}$
Input: Max epochs $E_{\max}$, learning rate η, batch size B
Output: Trained model parameters θ
foreach $(d_i, E_i, T_i) \in \mathcal{D}$ **do**
  $x_i \leftarrow \texttt{concat}(\text{``}\texttt{Queries: ''}, E_i, \text{``}\texttt{ Context: ''}, d_i)$
  $X_i \leftarrow \mathcal{T}_{\text{in}}(x_i)$; // Tokenize input
  $Y_i \leftarrow \mathcal{T}_{\text{out}}(T_i)$; // Tokenize target
for $e = 1$ **to** $E_{\max}$ **do**
  Sample mini-batch $\mathcal{B} = \{(X_i, Y_i)\}_{i=1}^{B}$
  foreach $(X_i, Y_i) \in \mathcal{B}$ **do**
    $\hat{Y}_i \leftarrow \mathcal{M}_\theta(X_i)$; // Generate output sequence
    $\mathcal{L}_i \leftarrow -\sum_{j=1}^{|Y_i|} \log P_\theta(y_{ij} \mid y_{i,<j}, X_i)$
  $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \left(\frac{1}{B} \sum_{i=1}^{B} \mathcal{L}_i \right)$
return θ

Source Dataset. We use the DBLP V14 collection as the foundation due to its rich metadata and broad coverage of computer science publications [34]. Our focus lies on publications from top-tier conferences and journals. Since no publicly available datasets provide verified ground-truth expert teams, prior work in team formation commonly adopts the assumption that co-authors of high-quality publications form effective teams [9, 16, 29–31]. We refine the original dataset D into a filtered subset D' using the function f defined as:

$$D' = f(D) = \{p_i \in D \mid v(p_i) \in V,\ |A(p_i)| \geq 2,\ 100 \leq |L(p_i)| \leq 5000\}$$

where $v(p_i)$ denotes the publication venue of paper p_i, V is the curated set of top-tier venues, $A(p_i)$ represents the set of distinct authors of p_i, and $L(p_i)$ indicates the length of its abstract (in characters). We retain only multi-author papers with non-trivial abstract content, which ensures both collaboration and sufficient textual context for prompt generation.

Author Expertise Augmentation. The DBLP V14 dataset does not provide explicit author-level expertise, which is essential for solving the team formation problem. While DBLP includes metadata such as paper keywords and fields of study [1], these attributes are associated with papers, not authors. Since effective team formation requires reasoning over the expertise of individuals, we augment the dataset with author-level expertise annotations.

To obtain these annotations, we employ `GPT-4` [25], to infer a ranked list of ten expertise areas from generic to specific $E(p_i)$ for each paper $p_i \in D'$, defined as: $g : D' \rightarrow E$. The generation process relies on carefully designed prompts that instruct the model to extract latent expertise based on available metadata. The system prompt used to guide `GPT-4` is shown below:

You are a helpful and honest assistant for labeling expertise required from authors to write academic research papers or publications. Expertise means the expert knowledge or skills the authors must have to write the paper or publication.

Table 1. Key statistics of constructed dataset.

Metric	Count
Number of Papers	26,051
Number of Unique Authors	21,911
Number of Collaborations among Authors	104,158
Avg Characters per Abstract	1,076

The paper or publication's title, abstract, keywords provided by authors, field of study labeled by others, and publication venue are provided. Please provide the top ten expertise areas required from authors to write it. The expertise you generate should be in order from generic to specific. All output should be in English. You must be at least 70% confident about the expertise and research topic. Otherwise, generate "NA".

This procedure yields a set of ten expertise labels for every paper in D', with no instance resulting in an "NA" response. The final dataset D'' contains the original abstract, author information, and the associated expertise annotations.

Cross-Model Validation of Expertise Augmentation. To assess the reliability of the expertise augmentation generated by `GPT-4`, we employed two additional large language models—`LLaMA 3.1` and `Qwen 2.5`—as independent validators.

Out of 26,051 annotated papers, only 27 (0.1%) received a "No" response from either model, and only 4 papers (0.015%) received a "No" from both. These results indicate high agreement across models and support the reliability of the `GPT-4`-generated expertise labels.

Key statistics of the final dataset D'' appear in Table 1.

Input Representations for Baseline Comparison. To assess the effectiveness of the proposed expertise annotations, we evaluate three additional input types commonly used in traditional team formation methods: `(1) TF-IDF` terms, `(2) Field of Study`, `(3) Keyword`. The Field of Study and Keyword metadata are provided in the DBLP V14 dataset. For the TF-IDF terms input, we construct term-based representations using the same textual sources used for expertise annotation: titles, abstracts, keywords, fields of study, and publication venues. We concatenate these sources, apply standard text preprocessing techniques—including lowercasing, tokenization, stopword removal, and stemming—and extract all unigrams, bigrams, and trigrams. We compute TF-IDF scores and retain the top 1,000 terms with the highest scores across the dataset. Table 2 summarizes the vocabulary size for each input type. For illustration, we include a concrete example from a paper titled *Correctness and parallelism in composite systems*. Figure 1 shows the distribution of the number of terms per paper for the three input types of the dataset.

5 Experiments and Discussions

Experimental Setup. We evaluate `PTF` using a train-test split with a test ratio 20% of the constructed dataset. To ensure that the model only predicts

Table 2. Vocabulary size and a representative example of input representations.

Type	Total	Example of One Paper's Content
TF-IDF	1,000	acm, architectur, assumpt, comput, comput scienc, correct, databas, degre, exist, extens, layer, make, nearest, nearest neighbor, neighbor, number, paper, parallel, principl, proceed, recent, scienc, search, semant, start, studi, symposium, theoret, theoret comput, theoret comput scienc, theori, transact, use, varieti, year
Field of Study	13,036	Computer science, Correctness, Composite number, Theoretical computer science, Nearest neighbor search
Keyword	116,237	composite system, nearest neighbor search
Expertise	39,636	Computer Science, Database Systems, Transaction Management, Parallel Computing, Concurrency Control, Distributed Systems, Theoretical Computer Science, Composite Systems, Correctness in Computing, Nearest Neighbor Search

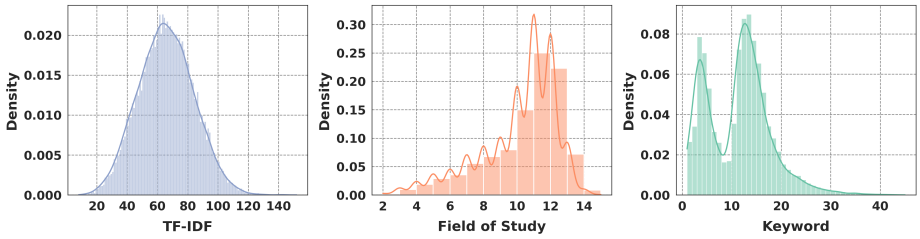

Fig. 1. Distribution of number of terms of TF-IDF, Field of Study and Keyword metadata in the dataset used for experiments.

author identifiers it has encountered during training, we restrict the test set to projects whose associated authors also appear in the training set, this procedure follows prior research such as [5, 26–31]. This constraint reflects a closed-world assumption and allows fair evaluation of the model's generalization to new project descriptions rather than unseen authors.

All experiments share the same train-test split, and all T5-based models are trained using identical hyperparameter configurations for consistency. We report performance using standard information retrieval metrics, including precision@10, recall@10, mean reciprocal rank (MRR), normalized discounted cumulative gain at rank 10 (NDCG@10), mean average precision (MAP), and F1 score, following prior work in neural team formation [6, 31, 35].

5.1 Performance Analysis

To examine how input and model capacity influences effectiveness, we consider three input variants: `(1) Expertise`, using only the inferred expertise annota-

Table 3. Comparison of PTF performance with different T5 model sizes using expertise and abstract as input. T5-Large achieves the best results, demonstrating the benefits of increased model capacity.

Input	Model	Precision	Recall	MRR	NDCG	MAP	F1
Expertise	PTF_s	0.0049	0.0165	0.0161	0.0129	0.0083	0.0163
	PTF_b	**0.0123**	**0.0392**	**0.0306**	**0.0296**	**0.0209**	**0.0412**
	PTF_l	0.0107	0.0334	0.0237	0.0248	0.0183	0.0364
Abstract	PTF_s	0.0056	0.0184	0.0180	0.0145	0.0093	0.0202
	PTF_b	0.0178	0.0546	0.0388	0.0409	0.0303	0.0626
	PTF_l	**0.0305**	**0.0930**	**0.0600**	**0.0690**	**0.0530**	**0.1024**
Expertise + Abstract	PTF_s	0.0059	0.0192	0.0189	0.0151	0.0096	0.0205
	PTF_b	0.0184	0.0585	0.0412	0.0434	0.0320	0.0651
	PTF_l	**0.0299**	**0.0928**	**0.0611**	**0.0687**	**0.0524**	**0.1016**

tions; (2) Abstract, using only the project abstracts; and (3) Expertise + Abstract, using both combined; and we evaluate the performance of PTF_s (uses T5-Small as base model), PTF_b (uses T5-Base as base model), and PTF_l (uses T5-Large as base model). The results are summarized in Table 3.

Comparing across three types of inputs, both Expertise + Abstract and Abstract-only as input representations provide substantial benefits over using structured expertise alone, indicating that full project descriptions provide the richest source of contextual information for team formation.

Expertise + Abstract achieves the best overall scores. However, it is noticeable that Abstract-only achieves comparably good performance, suggesting that adding expertise annotations adds minor redundancy or even noise to the input. These results reinforce the effectiveness of contextualized text input and highlight that each design choice in PTF—including the use of both structured and unstructured signals—offers measurable value, even if not strictly additive.

Looking into each of the three types of inputs, PTF_l achieves the highest scores across all evaluation metrics when using both Expertise + Abstract and Abstract-only as input, including more than a 50% relative gain in F1 score compared to PTF_b with Expertise + Abstract as prompt. This indicates that larger models are significantly more capable of capturing the semantic complexity of contextualized project descriptions and generating higher-quality expert team predictions. The smaller PTF_s and PTF_b variants offer more efficient alternatives when computational resources are limited. However, the substantial performance improvement of PTF_l highlights the scalability and headroom of prompt-based models when deployed with more powerful language model backbones. Interestingly, PTF_l does not outperform PTF_b when using Expertise as the sole input. We hypothesize this is due to the limited semantic richness in Expertise-only input, which may not sufficiently leverage the larger model capacity. This suggests that model size alone does not guarantee improved performance, particularly when input representations are sparse.

5.2 Baseline Comparison Across Input Representations

To the best of our knowledge, all existing approaches to the team formation problem rely on traditional term-based input representations. No prior work has incorporated contextualized textual descriptions or framed team formation as a prompt-based generation task. To enable a fair comparison, we evaluate our baseline model $\texttt{PTF}_b$ against existing methods using term-based input types.

First, we experiment with four types of term-based input representations: (1) TF-IDF, (2) Field of Study, (3) Keyword, and (4) expertise annotations. For methods that rely on Field of Study and Keyword metadata, we restrict the experiments to 20,915 papers and 20,209 papers respectively in our dataset that contain relative information. Despite this size reduction, we maintain the same train-test partitioning used across all experiments.

We compare our work with the following baseline methods: `Random`: a baseline that assigns experts to teams uniformly at random; this method does not rely on input features. `FNN`: a feed-forward neural network without Bayesian components, following the implementation in [6]. `BNN`: a Bayesian neural network proposed in [28], which incorporates uncertainty in expert selection. `Coherent`: a neural model introduced by [31] that jointly models collaboration likelihood and skill coverage to construct coherent expert teams. `Translative`: Transformer-based seq-to-seq model structure proposed by [35]. It is the current state-of-the-art.

Table 4 reports the performance of baseline methods and $\texttt{PTF}_b$ across the four term-based input types. Baseline models (`FNN`, `BNN`, `Coherent` and `Translative`) rely on term-based features, while $\texttt{PTF}_b$ processes the same inputs in a prompt-based seq-to-seq setting. Note that the baseline models' performance reported in Table 4 would not match with the models' original papers, due to each original paper's experiment setups are different.

Across all term-based input types, $\texttt{PTF}_b$ outperforms the baselines on Precision, Recall, and F1 score. The most significant improvements are observed with `expertise annotations`, where $\texttt{PTF}_b$ outperforms all other models by a wide margin. `Translative` model is the runner-up. It especially performs well for Keyword inputs. For Field of Study and Keywords inputs, $\texttt{PTF}_b$ gets higher Precision@10, Recall@10, whereas `Translative` model gets higher MRR. This indicates that the `Translative` model is good at ranking the first author, whereas $\texttt{PTF}_b$ retrieves more relevant authors overall. `FNN` model performs better than `BNN` model and `Coherent` model for TF-IDF inputs. `FNN` model is a non-Bayesian model comparing the other two models. It could be that the Bayesian component is not always beneficial.

Performance increases further using $\texttt{PTF}_l$ and combining expertise annotations with full project abstracts as input, demonstrating the additive value of natural language context. As `Translative` model uses a seq-to-seq model structure, besides expertise annotations alone, we also use expertise annotations combined with abstracts as input into this model. To realize it, we transform abstracts into chunked trigrams. However, adding abstracts as input makes the `Translative` model's performance drop. We think this is due to the `Translative` model treats the input as structured, clean skills, and the unstructured abstract likely dis-

Table 4. Performance of baseline models and $\texttt{PTF}_b$ across different term-based input types (TF-IDF terms, Field of Study, Keyword, and Expertise annotations) in comparison of $\texttt{PTF}_l$ and `Translative` models with Expertise + Abstract as input.

Input	Model	Precision	Recall	MRR	NDCG	MAP	F1
TF-IDF	`Random`	0.0003	0.0009	0.0016	0.0006	0.0007	0.0002
	`FNN`	0.0049	0.0136	0.0195	0.0100	0.0068	0.0000
	`BNN`	0.0003	0.0008	0.0019	0.0005	0.0008	0.0003
	`Coherent`	0.0001	0.0007	0.0015	0.0004	0.0007	0.0008
	`Translative`	0.0030	0.0084	0.0146	0.0084	0.0063	0.0000
	$\texttt{PTF}_b$	**0.0084**	**0.0275**	**0.0228**	**0.0208**	**0.0142**	**0.0282**
Field of Study	`Random`	0.0001	0.0004	0.0011	0.0002	0.0005	0.0003
	`FNN`	0.0000	0.0001	0.0009	0.0001	0.0005	0.0003
	`BNN`	0.0002	0.0005	0.0013	0.0002	0.0006	0.0003
	`Coherent`	0.0008	0.0000	0.0018	0.0005	0.0008	0.0009
	`Translative`	0.0059	0.0175	**0.0261**	0.0176	0.0144	0.0000
	$\texttt{PTF}_b$	**0.0094**	**0.0294**	0.0211	**0.0220**	**0.0162**	**0.0326**
Keyword	`Random`	0.0001	0.0005	0.0012	0.0002	0.0006	0.0003
	`FNN`	0.0001	0.0004	0.0012	0.0002	0.0006	0.0003
	`BNN`	0.0001	0.0002	0.0013	0.0001	0.0007	0.0004
	`Coherent`	0.0000	0.0005	0.0015	0.0003	0.0007	0.0010
	`Translative`	0.0076	0.0241	**0.0330**	**0.0239**	**0.0200**	0.0000
	$\texttt{PTF}_b$	**0.0093**	**0.0309**	0.0228	0.0232	0.0171	**0.0328**
Expertise	`Random`	0.0003	0.0009	0.0016	0.0006	0.0007	0.0002
	`FNN`	0.0002	0.0004	0.0263	0.0004	0.0007	0.0002
	`BNN`	0.0002	0.0007	0.0015	0.0004	0.0007	0.0003
	`Coherent`	0.0001	0.0007	0.0013	0.0004	0.0006	0.0008
	`Translative`	0.0054	0.0170	0.0245	0.0168	0.0136	0.0000
	$\texttt{PTF}_b$	**0.0123**	**0.0392**	**0.0306**	**0.0296**	**0.0209**	**0.0412**
Expertise+Abstract	`Translative`	0.0037	0.0110	0.0163	0.0109	0.0089	0.0000
	$\texttt{PTF}_l$	**0.0299**	**0.0928**	**0.0611**	**0.0687**	**0.0524**	**0.1016**

tracted the model. The attention probably got spread over too many irrelevant inputs. This shows the `Translative` model's limitation for long text inputs.

These results emphasize a fundamental limitation in the existing team formation models: although they can operate on structured inputs, they fail to exploit the latent semantic cues present in descriptive project text. In contrast, prompt-based model `PTF` effectively leverage both structured and unstructured input, enabling more accurate and context-aware expert team prediction.

5.3 Sensitivity Analysis

To evaluate how the number of input expertise terms affects model performance, we conduct a sensitivity analysis for the `PTF` models. This experiment tests the model's robustness to varying degrees of structured input and aims to quantify the incremental benefit of additional expertise annotations. All experiments use the same architecture and hyperparameter settings to ensure comparability.

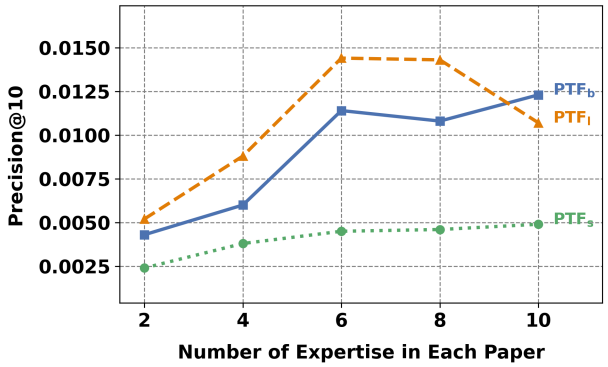

Fig. 2. Sensitivity analysis of model performance with varying numbers of expertise.

We vary the number of expertise terms from 2 to 10, incrementing by two. Figure 2 shows that increasing the number of expertise terms generally leads to better performance. For $\texttt{PTF}_b$ and $\texttt{PTF}_l$ models, the largest gains occur between 2 and 6 terms, after which the performance improvements of $\texttt{PTF}_b$ taper off and the performance of $\texttt{PTF}_l$ decreases. This suggests diminishing returns from additional expertise information beyond a certain threshold.

These results indicate that while a small number of structured expertise signals can significantly enhance model performance, richer input provides more stable and accurate team predictions. However, they also highlight that the models can perform reasonably well with partial input, which supports their potential deployment in settings where complete expertise profiles may be unavailable.

5.4 Findings Summary

Our experimental results reveal several key insights into the effectiveness and limitations of the proposed prompt-driven team formation framework. *First*, the combination of a seq-to-seq model such as T5 with contextualized text input leads to substantial performance gains. This confirms the value of leveraging detailed project descriptions to infer latent skill requirements, something that traditional keyword- or metadata-driven methods cannot capture effectively. *Second*, while structured expertise annotations alone underperform relative to full abstracts, they still offer a compact and informative representation of author capabilities. In settings where contextualized input is unavailable, these annotations serve as a viable alternative and can complement free-form text when combined. However, our experiment results suggest that their additive value may be limited, potentially due to noise introduced during automatic annotation. *Third*, model capacity also plays a critical role in maximizing the benefits of contextual input. Larger variants such as $\texttt{PTF}_l$ demonstrate a clear advantage over smaller models, indicating that prompt-based generation relies heavily on the model's expressive power to align expertise with project semantics. As a cost, fine-tuning the $\texttt{PTF}_l$ models consume more computation resources. *Fourth*, an important but underexplored factor in this work is the role of generation-time

hyperparameters—such as beam size, top-k sampling, diversity penalty, and repetition penalty—which influence the diversity and quality of the predicted expert teams. This study maintains consistent decoding settings across all experiments to ensure comparability, but future work could systematically investigate how these parameters affect team composition, especially under constraints like diversity, novelty, or fairness. *Fifth*, it is worth noting that the performance of all team formation models evaluated, including PTF, remains noticeably low. We attribute this in part to the limitation in the current evaluation setup, which could result in systematically underestimated model performance. Ideally, a proper evaluation of the team formation problem would require a dataset that provides ranking scores for all possible combinations of candidate team members for each task. However, to the best of our knowledge, no such real-world dataset exists. Consequently, existing studies on team formation typically rely on datasets such as DBLP as a practical, albeit limited, alternative. While we attempt to mitigate this limitation by restricting our dataset to top-tier publications, the underlying issue remains. This highlights significant opportunities for future research to improve the study of team formation, particularly in terms of data sources, problem formulation, and experimental design. Overall, our findings validate the core intuition behind `PTF`: contextual prompts combined with fine-tuned sequence models offer a powerful new paradigm for expert team formation. Yet the effectiveness of this approach depends on high-quality input representations, appropriate model capacity, and potentially, a more adaptive decoding strategy.

6 Conclusion

We proposed a framework for contextual prompt-driven team formation that effectively utilizes contextualized text inputs. By fine-tuning a T5-Large model on a novel enriched dataset, our model shows substantial improvements over traditional techniques. The findings highlight the importance of comprehensive project descriptions and expertise representation in enhancing team formation outcomes. This work lays the groundwork for future research into more effective and scalable team formation solutions. In the future, we plan to add more datasets from other domains and prepare a unique benchmark Test Set so that all models can compare based on that.

References

1. AMiner: Aminer citation (2023), https://www.aminer.cn/citation
2. Baykasoglu, A., Dereli, T., Das, S.: Project team selection using fuzzy optimization approach. Cybernetics and Systems: An International Journal **38**(2), 155–185 (2007)
3. Brown, T., Mann, B., Ryder, N., et al.: Language models are few-shot learners. Advances in Neural Information Processing Systems **33**, 1877–1901 (2020)
4. Bryson, S., Davoudi, H., Golab, L., Kargar, M., Lytvyn, Y., Mierzejewski, P., Szlichta, J., Zihayat, M.: Robust keyword search in large attributed graphs. Information Retrieval Journal **23**, 502–524 (2020)

5. Dara, M., Rad, R.H., Zarrinkalam, F., Bagheri, E.: Retrieval-augmented neural team formation. In: European Conference on Information Retrieval. pp. 362–371. Springer (2025)
6. Dashti, A., Saxena, K., Patel, D., Fani, H.: Opentf: a benchmark library for neural team formation. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 3913–3917 (2022)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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, volume 1 (long and short papers). pp. 4171–4186 (2019)
8. Esgario, J.G., da Silva, I.E., Krohling, R.A.: Application of genetic algorithms to the multiple team formation problem. arXiv preprint arXiv:1903.03523 (2019)
9. Fani, H., Barzegar, R., Dashti, A., Saeedi, M.: A streaming approach to neural team formation training. In: European Conference on Information Retrieval. pp. 325–340. Springer (2024)
10. Fani, H., Jiang, E., Bagheri, E., Al-Obeidat, F., Du, W., Kargar, M.: User community detection via embedding of social network structure and temporal content. Information Processing & Management **57**(2), 102056 (2020)
11. Fitzpatrick, E.L., Askin, R.G.: Forming effective worker teams with multi-functional skill requirements. Computers & industrial engineering **48**(3), 593–608 (2005)
12. Golzadeh, K., Golab, L., Szlichta, J.: Explaining expert search and team formation systems with exes. arXiv preprint arXiv:2405.12881 (2024)
13. Hamidi Rad, R., Cucerzan, S., Chandrasekaran, N., Gamon, M.: Interactive topic tagging in community question answering platforms. In: European conference on information retrieval. pp. 195–209. Springer (2024)
14. Kargar, M., An, A.: Discovering top-k teams of experts with/without a leader in social networks. In: Proceedings of the 20th ACM international conference on Information and knowledge management. pp. 985–994 (2011)
15. Kargar, M., Golab, L., Srivastava, D., Szlichta, J., Zihayat, M.: Effective keyword search over weighted graphs. IEEE Transactions on Knowledge and Data Engineering **34**(2), 601–616 (2020)
16. Kargar, M., Zihayat, M., An, A.: Finding affordable and collaborative teams from a network of experts. In: Proceedings of the 2013 SIAM international conference on data mining. pp. 587–595. SIAM (2013)
17. Kaw, S., Kobti, Z., Selvarajah, K.: Transfer learning with graph attention networks for team recommendation. In: 2023 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2023)
18. Khan, A., Golab, L., Kargar, M., Szlichta, J., Zihayat, M.: Compact group discovery in attributed graphs and social networks. Information Processing & Management **57**(2), 102054 (2020)
19. Lappas, T., Liu, K., Terzi, E.: Finding a team of experts in social networks. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 467–476 (2009)
20. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in neural information processing systems **33**, 9459–9474 (2020)
21. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)

22. Nguyen, H., Hamidi Rad, R., Bagheri, E.: Pydhnet: a python library for dynamic heterogeneous network representation learning and evaluation. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 4936–4940 (2022)
23. Nguyen, H., Rad, R.H., Zarrinkalam, F., Bagheri, E.: Dyhnet: Learning dynamic heterogeneous network representations. Information Sciences **646**, 119371 (2023)
24. Nikzad-Khasmakhi, N., Balafar, M., Feizi-Derakhshi, M.R., Motamed, C.: Exem: Expert embedding using dominating set theory with deep learning approaches. Expert Systems with Applications **177**, 114913 (2021)
25. OpenAI: Gpt-4 (2023), https://openai.com/index/gpt-4-research/
26. Rad, R.H., Bagheri, E., Kargar, M., Srivastava, D., Szlichta, J.: Retrieving skill-based teams from collaboration networks. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. pp. 2015–2019 (2021)
27. Rad, R.H., Bagheri, E., Kargar, M., Srivastava, D., Szlichta, J.: Subgraph representation learning for team mining. In: Proceedings of the 14th ACM Web Science Conference 2022. pp. 148–153 (2022)
28. Rad, R.H., Fani, H., Bagheri, E., Kargar, M., Srivastava, D., Szlichta, J.: A variational neural architecture for skill-based team formation. ACM Transactions on Information Systems **42**(1), 1–28 (2023)
29. Rad, R.H., Fani, H., Kargar, M., Szlichta, J., Bagheri, E.: Learning to form skill-based teams of experts. In: Proceedings of the 29th ACM international conference on information & knowledge management. pp. 2049–2052 (2020)
30. Rad, R.H., Nguyen, H., Al-Obeidat, F., Bagheri, E., Kargar, M., Srivastava, D., Szlichta, J., Zarrinkalam, F.: Learning heterogeneous subgraph representations for team discovery. Information Retrieval Journal **26**(1), 8 (2023)
31. Rad, R.H., Seyedsalehi, S., Kargar, M., Zihayat, M., Bagheri, E.: A neural approach to forming coherent teams in collaboration networks. In: EDBT. pp. 2–440 (2022)
32. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research **21**(140), 1–67 (2020)
33. Sapienza, A., Goyal, P., Ferrara, E.: Deep neural networks for optimal team composition. Frontiers in big Data **2**, 14 (2019)
34. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 990–998 (2008)
35. Thang, K., Hosseini, H., Fani, H.: Translative neural team recommendation: From multilabel classification to sequence prediction. In: Proceedings of the 48th International ACM SIGIR (2025), to appear
36. Tian, Y., Song, H., Wang, Z., Wang, H., Hu, Z., Wang, F., Chawla, N.V., Xu, P.: Graph neural prompting with large language models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 19080–19088 (2024)
37. Wi, H., Oh, S., Mun, J., Jung, M.: A team formation model based on knowledge and collaboration. Expert Systems with Applications **36**(5), 9121–9134 (2009)
38. Yang, Y., Xia, L., Luo, D., Lin, K., Huang, C.: Graphpro: Graph pre-training and prompt learning for recommendation. In: Proceedings of the ACM on Web Conference 2024. pp. 3690–3699 (2024)
39. Zihayat, M., An, A., Golab, L., Kargar, M., Szlichta, J.: Authority-based team discovery in social networks. arXiv preprint arXiv:1611.02992 (2016)