Promoting Research Collaboration with Open Data Driven Team Recommendation in Response to Call for Proposals

Siva Likitha Valluru[1], **Biplav Srivastava**[1], **Sai Teja Paladi**[1], **Siwen Yan**[2], **Sriraam Natarajan**[2]

[1]Artificial Intelligence Institute, University of South Carolina
[2] Statistical Artificial Intelligence and Relational Learning Group (StARLinG Lab), University of Texas
{svalluru@email., biplav.s@, spaladi@email.}sc.edu, {siwen.yan, sriraam.natarajan}@utdallas.edu

Abstract

Building teams and promoting collaboration are two very common business activities. An example of these are seen in the *TeamingForFunding* problem, where research institutions and researchers are interested to identify collaborative opportunities when applying to funding agencies in response to latter's calls for proposals. We describe a novel *deployed* system to recommend teams using a variety of AI methods, such that (1) each team achieves the highest possible skill coverage that is demanded by the opportunity, and (2) the workload of distributing the opportunities is balanced amongst the candidate members. We address these questions by extracting skills latent in open data of proposal calls (demand) and researcher profiles (supply), normalizing them using taxonomies, and creating efficient algorithms that match demand to supply. We create teams to maximize goodness along a novel metric balancing short- and long-term objectives. We validate the success of our algorithms (1) quantitatively, by evaluating the recommended teams using a goodness score and find that more informed methods lead to recommendations of smaller number of teams but higher goodness, and (2) qualitatively, by conducting a large-scale user study at a college-wide level, and demonstrate that users overall found the tool very useful and relevant. Lastly, we evaluate our system in two diverse settings in US and India (of researchers and proposal calls) to establish generality of our approach, and deploy it at a major US university for routine use.

Introduction

In the recent decade, there has been an increased interest in studying teamwork skills and their impacts in a multitude of domains (e.g., academia (Alberola et al. 2016), social networking (Anagnostopoulos et al. 2012), project management (Noll et al. 2016), healthcare (Nawaz et al. 2014)). Building successful teams is a common business strategy (e.g., forming rescue and relief teams in response to an emergency (Gunn and Anderson 2015), establishing entrepreneurial teams for new ventures (Lazar et al. 2020), forming teams dynamically in context of multi-agent systems (e.g., supply chains) (Gaston and desJardins 2005)). In this paper, we focus on teaming for researchers applying to funding agencies in response to their call for proposals, using group recommendation strategies. The advantage of this setting is that the required data is already publicly available. A large amount of research funding in public universities comes from external funding agencies such as National Science Foundation (NSF) and National Institutes of Health (NIH). These opportunities often require multi-disciplinary teams from a wide variety of backgrounds to be quickly assembled. However, not every single team is often successful in achieving their goals, due to factors such as lack of accurate information, time, or communication, and incompatibility in terms of skill sets among team members.

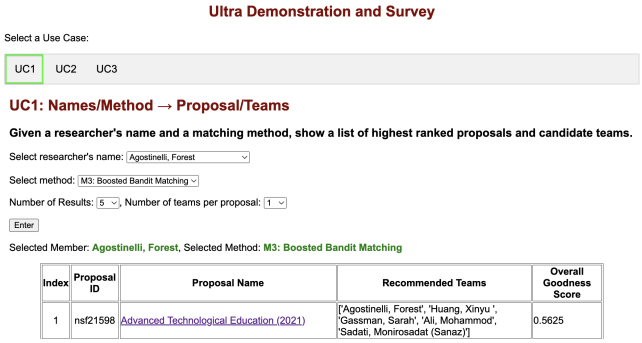

Figure 1: A demo use case, UC_1, showing a team participant view of ULTRA.

We build upon prior work, ULTRA (**U**niversity-**L**ead **T**eam Builder from **R**FPs and **A**nalysis) (Srivastava et al. 2022), a novel AI-based prototype for assisting with team formation when researchers respond to calls for proposals from funding agencies. In this paper, we interchangeably use the term *call for proposal* with *request for proposal (RFP).* Figure 1 shows a demo[1] view of how the system works for an individual user who can become a team participant. The system first extracts technical skills from proposal calls found at publicly available data sources (e.g., NSF archives) and those present in online profiles of researchers (e.g., personal homepages, Google Scholar history), along with any additional teaming constraints that administrators or team partic-

[1]A full demo interaction with the ULTRA system can be found at https://www.youtube.com/watch?v=8MUtxsfVNIU. The tool is deployed at http://casy.cse.sc.edu/ultra/teaming/. Additional details about usecases, experiments, and survey resources are at (Valluru, Srivastava, and Paladi 2023).

ipants of an institution may provide. Using AI and NLP techniques, ULTRA next provides a plausible list of candidate teams for each proposal call, where each team has at least two members. Our prior work (Srivastava et al. 2022), however, is a use case of a *sequential* single-item recommendation problem, where solutions are often limited by known issues such as cold start (Abdollahpouri, Burke, and Mobasher 2019) or popularity bias (Yalcin and Bilge 2022). Therefore, we expand on this work to include group recommendation and novel AI methods to recommend optimal teams.

Our contributions in the paper are that we:

- formulate a novel *group recommendation problem* to promote research collaborations where the objective is to suggest teams that could respond to calls for proposals, using skills found in open data, and balancing short- and long-term optimization.
- introduce a metric to measure goodness of teams and consider a configurable set of criteria: redundancy, group (set) size, coverage, and robustness.
- solve the novel problem using a variety of AI methods: string, taxonomy, and bandit (relational learning) methods, and compare them with a randomized baseline.
- establish the benefit of the solution methods quantitatively using goodness metric. We find that more *informed methods lead to recommendations of smaller number of teams but higher goodness*.
- establish the benefit of the system qualitatively using an IRB-approved preliminary survey at a College of a major US University. We find that *users broadly consider the tool useful as well as relevant* but more studies are needed.
- demonstrate the generality of the approach with experiments at two different institutions in US and India.
- create and publish a teaming dataset that is available for research.

Related Work

A well-studied problem of AI in team formation is the *Hedonic Games* framework (Aziz et al. 2017; Gairing and Savani 2010), a coalition structure consisting of disjoint coalitions that cover all players, where each player estimates a valuation for other players in their group. Team formation has also been considered in project management, where evaluation of team members is conducted by measuring attributes such as leadership skills, technical talent, problem solving capabilities, and cultural relevance (Warhuus et al. 2021). Sports leagues assess the physical and functional well-being of players before forming teams (Dadelo et al. 2014). Such a task can be complex as teams often require players who are efficient in multiple roles. (Ahmed, Deb, and Jindal 2013) used the NSGA-II algorithm and employed an evolutionary multi-objective approach to obtain a high-performing team for cricket tournaments. Some factors that the system takes into consideration are batting average and bowling performance of individual players, wicket-keeper's past performances, and other rule-based constraints. Depending on the team selection strategy, each of the above factors are ranked (or nulled) according to importance and the final solution set is obtained according to it. If all the factors are deemed equally important, a domination approach is applied instead, i.e., sorting teams according to non-dominating factors (e.g., based on total cost of team formation) and picking the best front solution (i.e., lowest cost team).

Closer to our setting, (Machado and Stefanidis 2019) proposed brute force and heuristic approaches to create team recommendations in multidisciplinary projects, where most suitable candidates are incrementally selected until project requirements are fulfilled. However, unlike our work, they do not use any metrics to quantitatively measure the effectiveness of recommended teams, and additionally assume that an individual member may completely fulfill one skill requirement. The complexity of making multi-criteria decisions and building successful teams also increases with the number of available candidates. Evaluating the goodness of every team permutation quickly becomes computationally prohibitive and NP-hard (Roy et al. 2014). The semantics of group recommendation algorithms were also defined by a means of a Consensus Function, where *disagreement* amongst group members was introduced as a factor to influence the generated recommendations (Amer-Yahia et al. 2009). Alternatively, methods have also been proposed to seek a certain level of *agreement* amongst group members, where individuals preferences are iteratively brought closer, until a desired threshold is achieved (Castro et al. 2015). Search-based optimization techniques were also explored using a queuing simulation model to maximize resource usage in software project management (Di Penta, Harman, and Antoniol 2011). Another application of team formation is in entrepreneurial domains (Lazar et al. 2020), where entrepreneurs search for trustworthy partners *and* investors when building new ventures.

Existing literature systematically answers many questions regarding what, why, and how teams are formed (Costa et al. 2020; Juárez, Santos, and Brizuela 2021). However, recruiting a group of experts to work towards a common goal does not guarantee that they will *always* operate as a team.

Drawbacks of Single-Item Recommendation. A common objective of many recommender algorithms (Su and Khoshgoftaar 2009; Cremonesi, Koren, and Turrin 2010) is to learn each individual's preferences through his interactions with a system, estimate his satisfaction for items he has not interacted with before, and return the top-k items with highest estimated ratings. However, many single-item recommender systems suffer from known issues such as popularity bias, where suggestions show an uneven distribution in recommending similar items, and the cold start problem, a state where a new user wishes to interact with a system but there is not enough information about his preferences to make an accurate decision.

Motivation. We therefore expand on our prior work, ULTRA (**U**niversity-**L**ead **T**eam Builder from **R**FPs and **A**nalysis) (Srivastava et al. 2022), and consider a group recommendation setting that promotes research collaboration using novel AI methods to recommend optimal teaming suggestions. Our work encourages multi-functional and interdisciplinary teams to form, and brings together members

from various disciplines of work responsibility.

Problem Setting

Problem Formulation

In this section, we introduce the domain of our work and provide the groundwork for the problem. Funding agencies periodically announce Requests For Proposals (RFPs) on specific themes where they are looking for ideas to fund. Researchers in turn respond to those calls, with proposals, where they explain their ideas and detail a meaningful and actionable plan for how they plan to achieve said goals within a specific allotted budget, time frame, and other constraints. In doing so, they also often look to team with other colleagues to respond to such calls. As such, we consider a teaming environment where the availability of candidate participants may change at any given time, often sporadically.

In terms of terminology, let S denote the set of all skills. Then, the *demand* for teaming is represented by the set of skills S_i desired by an RFP c_i. Further, the *supply* is represented at an institution by the set of researchers R along with their respective research interests and profiles to satisfy the demand. Teaming objectives may be short-term (ST), long-term (LT), or a combination of the two. An example of a short-term goal would be to meet the immediate requirements and skill needs, $S = \{s_1, s_2, ..., s_\alpha\}$ set forth by a call for proposal c_i. Each call in $C = \{c_1, c_2, ..., c_M\}$ requires a very specific skill set S_i and an immediate and optimal assembling of available candidate researchers, $R = \{r_1, r_2, ..., r_N\}$. We then solve our teaming setup in three phases: (1) we first *match* all researchers who may be of interest to the calls based on skills needed, (2) *group* which subset of researchers should be recommended to be in a team t_i, and (3) compute goodness score g_i for each team in $T = \{t_1, t_2, ..., t_\beta\}$ and recommend the top-*k* suggestions to interested users.

Similarly, some example scenarios of a long-term objective would be to maximize the number of funded awards A given to a researcher r_j over a time period (LT_t^A), have a robust (diversified) pool of experienced talent (LT_t^R), and satisfy diversity goals of researchers' institutions.

Our system would be of interest to at least two types of users: (1) administrators at researchers' organizations (e.g., university institutions) who want to promote more collaborations, proposals, and diversity at their institutions, and (2) potential team members who will respond to a given RFP c_i and are looking for collaborative opportunities. Various environments call for different teams to be formed and matched with relevant opportunities. The candidate member set also may change over time, along with each of their skills and research interests.

Each user (i.e., admin or researcher) will interact with the system when a call for proposal c_i is announced. Based on the requirements set by c_i, along with profiles extracted from researchers, the system will then algorithmically suggest teaming choices $T = \{t_1, t_2, ..., t_\beta\}$, which the users may accept or reject. We evaluate the efficacy of our algorithms and validate the teaming outputs via a goodness score.

Use Cases

We show at least three practical use cases (UCs) to demonstrate the utility of our system. Each use case has various input prompts to select from and includes different algorithms that can be used to recommend or suggest proposals and teams to interested users.

For the first use case, UC_1, given a researcher's name and a teaming method M_i, we display a list of k highest ranked proposals and possible teams (shown in Figure 1). Similarly, for UC_2, given a proposal call c_i from a list of recently announced proposals C (ideally refreshed in real-time or regularly), we display the best possible teams T available for c_i. And the final use case, UC_3, takes input in the form of a research interest and teaming method, and displays respective matching proposals and teams for those parameters. We empirically evaluate the three use cases and the functionality of the four methods used in the later sections of the paper.

System Design

In this section, we describe the general architecture of our system and its most important components.

Metrics

A challenge in team recommendation scenarios is how to adapt to a group as a whole, given individual preferences of each member within a group (Boratto et al. 2010). Literature on team collaboration emphasizes that teams be organized to ensure diversity of team members (He, von Krogh, and Sirén 2022) Additionally, when it comes to the relationship between collaboration and scientific impact, team size also matters. While equally dependent on the number of requirements and time constraints set forth by an RFP, evidence also suggests that shorter teams also yield quick outputs, as they allow for higher accountability, autonomy, and flexibility amongst team members (Trope 2023).

In traditional literature on recommendation, there are metrics to measure the position of a (single correct) result (e.g., mean reciprocal rank (MRR) and top-*k* for ranking). However, our focus is on metrics that reflect the goodness of *multiple* good results (i.e., team recommendations). Still, positional metrics are important since they reflect users' acceptance of the results. We have implicitly incorporated them by displaying teaming results in descending order of the team goodness score. Therefore, by incorporating both considerations (team quality and user acceptance), for each candidate team t_i, we measure its effectiveness towards c_i using a goodness score g_i. The score denotes the chances of success for a team to fulfill the requirements of c_i, and is computed by taking the weighted mean of four configurable metrics: *redundancy, set size, coverage,* and *k-robustness.* We now explain the metrics used, along with how goodness is calculated for a team.

Redundancy. This is defined as the percentage of demanded skills that are commonly shared amongst multiple researchers. A trade-off that arises with skill redundancy is

team robustness versus diversity. While an increased redundancy ensures expert competence within a group of skills, it also risks limiting the amount of skill diversity that a team has.

Set Size. This metric is defined as the size of the candidate team. A trade-off present here is skill coverage and robustness versus funding amount split per researcher. A smaller team size runs the risk of not being able to accomplish the necessary goals of the proposal call (e.g., due to unavailability of any team members, longer efforts needed to complete a task), whereas a larger one mitigates that risk but lowers the amount of funding that each researcher receives. In addition, large teams lead to other disadvantages such as unequal participation amongst members, longer time needed to make decisions (e.g., due to intrinsic conflicts or lack of timely communication), whereas a smaller team fosters for accountability, individuality, and flexibility in ideas and schedules (McLeod, Lobel, and Cox Jr 1996).

Coverage. This metric represents the percentage of proposal-required skills that are satisfied by the candidate team as a whole. A candidate member's skills are defined from those listed within their personal webpages and extracted from other research profiles such as Google Scholar. We devise this metric by borrowing the idea that diverse expertise often invites individuals with different perspectives but also may lack common shared experience (He, von Krogh, and Sirén 2022). A mix of team members with diverse knowledge, skill sets, and abilities have also been thought to bring forth their unique skills to the team and provide it with the broadest possible skill set.

***k*-Robustness.** We borrow this metric from (Okimoto et al. 2015), where each team needs to be able to equally satisfy the teaming constraints even after the removal or unavailability of *k* researchers. Such a team is defined as *k*-robust.

Goodness Score. We first normalize the aforementioned metrics to make their values query-independent. Next, we assign each of the metrics a predefined weight by default. Given our use cases, the weights are defined by the intuition to yield the maximum profit (i.e., project completion) and credibility (i.e., project quality) a team can achieve. A high coverage and robustness are therefore more desired for overall project success, whereas high redundancy and set size are less prioritized. As a result, for each candidate team t_i, we penalize the latter two metrics and reward the former. The penalized metrics are set to a negative weight of -1, whereas the desired ones are set to a positive weight of $+1$. Finally, the goodness score for a team is calculated using the weighted mean from all four metrics. The diversity of metrics increases the potential to provide strength and resilience to the overall model. For additional reference, we make our metrics tool publicly available on GitHub [2].

Methods

M0: Random Team Formation. We consider random team selection as our baseline, where candidate teams are formed in a randomized manner, without any adherence to the skills in demand, and matched to an arbitrary proposal, regardless of relevance. Given any c_i, we select a random number of individuals from a pool of N available researchers to form teams T. Using our goodness function, we next evaluate each team t_j by extracting the skills S_i required by the proposal c_i and checking how many are solvable by the team at hand. Algorithm 1 provides pseudocode for *M0*.

Algorithm 1: *M0:* Random Team Formation

Input: Calls $C = \{c_1, \ldots, c_M\}$, Researchers $R = \{r_1, \ldots, r_N\}$
Output: Teams $T = \{t_1, \ldots, t_\beta\}$, good. scores $G = \{g_1, \ldots, g_\beta\}$

for $c_i \in C$ **do**
  $T = \{\}, G = \{\}$
  for $j = 1, 2, \ldots, \beta$ **do**
    Let t_j be a *k* random sampling of N researchers.
    $T = T \bigcup \{t_j\}$
    Compute goodness score g_j for team t_j.
    $G = G \bigcup \{g_j\}$
  end for
end for

M1: Team Formation Using String Matching. Given a call for proposal c_i, we extract the technical skills required from it using its title and synopsis as inputs. We remove stop words and delimiters and use keyword extraction to gather only the relevant skills needed in S_i. Similarly, we gather the research interests each available researcher r_j has listed on their personal webpage and demonstrated with their Google Scholar history. We denote this as $\sigma(r_j)$. We then check if there are any common interests between $\sigma(r_j)$ and S_i.

Algorithm 2: *M1:* Team Formation Using String Matching

Input: Calls $C = \{c_1, \ldots, c_M\}$, Researchers $R = \{r_1, \ldots, r_N\}$, string matching threshold th_{M1}
Output: Teams $T = \{t_1, \ldots, t_\beta\}$, good. scores $G = \{g_1, \ldots, g_\beta\}$

for $c_i \in C$ **do**
  $T = \{\}, G = \{\}$
  Extract technical skills $S_i = \{s_1, s_2, \ldots, s_\alpha\}$ for each c_i.
  Initialize $candidate\_researchers = []$.
  for $s \in S$ **do**
    for $j = 1, 2, \ldots, N$ **do**
      if s is in $\sigma(r_j)$ by satisfying th_{M1} **then**
        Add r_j to $candidate\_researchers$ [].
      end if
    end for
  end for
  Using $candidate\_researchers$ [], form each team t_k, prioritizing members with highest string matches.
  $T = T \bigcup \{t_k\}$
  Compute goodness score g_k for each team t_k.
  $G = G \bigcup \{g_k\}$
end for

[2]Metrics tool: https://github.com/ai4society/Ultra-Metric

Algorithm 3: *M2:* Team Formation Using Taxonomical Matching

Input: Calls $C = \{c_1, \ldots, c_M\}$, Researchers $R = \{r_1, \ldots, r_N\}$, string matching threshold th_{M2}
Output: Teams $T = \{t_1, \ldots, t_\beta\}$, good. scores $G = \{g_1, \ldots, g_\beta\}$

for $c_i \in C$ **do**
  $T = \{\}$, $G = \{\}$
  Extract technical skills $S_i = \{s_1, s_2, \ldots, s_\alpha\}$ for each c_i.
  Calculate *n-grams* from c_i, and add to S_i.
  Using th_{M2}, map each $s \in S_i$ with relevant classification codes from ACM-CCS, i.e., $\delta(S_i)$.
  Initialize $candidate\_researchers = []$.
  for $j = 1, 2, \ldots, N$ **do**
    Using the same th_{M2}, calculate $\delta(\sigma(r_j)$.
    if $\delta(S_i) \bigcap \delta(\sigma(r_j)! = \emptyset$ **then**
      Add r_j to $candidate\_researchers$ [].
    end if
  end for
  Using $candidate\_researchers$ [] and for each r_j, form each team t_k, prioritizing members with highest taxonomical matches.
  $T = T \bigcup \{t_k\}$
  Compute goodness scores g_k for each team t_k.
  $G = G \bigcup \{g_k\}$
end for

Given a pattern string of length *x* and a target string of length *y*, we determine if there is any overlap between those two using a string-match threshold th_{M1}. Algorithm 2 provides pseudocode for *M1*.

M2: Team Formation Using Taxonomical Matching. We further improve the accuracy and precision of the previous methods, *M0* and *M1*, by considering query-based *semantic matching*, combined with the use of a *taxonomy*. We use a poly-hierarchical, subject-based ontology, provided by the *ACM Computing Classification System (CCS)* (ACM 2012). There are over two thousand topics listed that broadly reflect the research areas pursued in the computing discipline. These are further organized into categories and concepts, with up to four branches of structure. We use this ontology to determine if two research skills may be matched semantically rather than only string-wise. For instance, if two researchers, r_a and r_b, each had the respective skills, *"natural language processing"* and *"knowledge representation"*, the method *M1* will return an extremely low string-match score and deny any association between the terms. However, using ACM-CCS, *M2* will categorize these two interests under *"artificial intelligence"* and consider the possibility that r_a and r_b may belong within the same team for a call c_i.

Given a call for proposal c_i, we extract the relevant skills needed, S_i. For each skill in S_i, we then use *n-grams* to compare each sequence of words with the concepts in ACM-CCS using a string match threshold th_{M2}. We denote this step with $\delta(S_i)$. Each concept is mapped to a specific code, and we use those codes to search for candidate researchers. For each available researcher r_j, we first extract their research interests $\sigma(r_j)$. For each interest, we similarly calculate $\delta(\sigma(r_j))$ to get the relevant codes from ACM-CCS. Finally, we form teams and calculate goodness based on a match between the codes. Algorithm 3 provides pseudocode for *M2*.

M3: Team Formation Using Boosted Bandit. According to requirements of proposals and expertise of researchers, the previous three methods apply manually-crafted rules to matching researchers to teams. However, *M3* extracts rules automatically from data. With more facts and data provided, this method is able to learn more complex rules automatically. We consider the strategy taken by (Kakadiya, Natarajan, and Ravindran 2021), and formulate the problem as team recommendation using contextual bandits. The key idea is that given the skills required by proposals and the research interests/expertise (denoted as $\mathbf{x}$), the goal is to

Algorithm 4: *M3:* Team Formation Using Boosted Bandit

Input: Calls $C = \{c_1, \ldots, c_M\}$, Researchers $R = \{r_1, \ldots, r_N\}$
Output: Teams $T = \{t_1, \ldots, t_\beta\}$, good. scores $G = \{g_1, \ldots, g_\beta\}$

function BOOSTEDTREES($predicates$)
  for $p \in predicates$ **do**
    Let I be a pre-set number of iterations.
    for $i = 1, 2, \ldots, I$ **do**
      Generate examples for the regression-tree learner: GENERATEEX($p, predicates, F^p_{i-1}$)
      Get new regression tree, which approximates functional gradient $\triangle_i(p)$, and update current model F^p_i.
    end for
    Get final potential: $\psi_I = \psi_0 + \triangle_1(p) + \ldots + \triangle_I(p)$
  end for
  return
end function
function GENERATEEX($p, predicates, F$)
  Initialize examples $E = \emptyset$.
  for $j = 1, 2, \ldots, X_p$ **do**
    Calculate probability of predicate p being true.
    Compute gradient and update regression examples.
    Compute regression values based on the groundings of current example.
  end for
  return regression examples E.
end function
for $c_i \in C$ **do**
  $T = \{\}$, $G = \{\}$
  Extract technical skills $S_i = \{s_1, s_2, \ldots, s_\alpha\}$ for each c_i.
  Initialize $candidate\_researchers = []$.
  Initialize $predicates[]$.
  Show associations between (c_i, S_i), and (r_j, $\sigma(r_j)$) as $predicates[]$.
  Get candidate researchers from BOOSTEDTREES($predicates$) and add to $candidate\_researchers$ []. Form each team t_k, prioritizing members with highest probability matches.
  $T = T \bigcup \{t_k\}$
  Compute goodness score g_k for each team t_k.
  $G = G \bigcup \{g_k\}$
end for

learn $P(y \mid \mathbf{x})$ where y is whether a researcher is a potential candidate for the proposal. This is to say that y is a two-argument predicate $candidate(r_j, c_i)$, which states that the researcher r_j is a candidate for the proposal c_i. The key idea in boosted bandit is to represent this as a sigmoid, $P(y \mid \mathbf{x}) = \frac{e^{\psi(y|\mathbf{x})}}{\sum_{y'} e^{\psi(y'|\mathbf{x})}}$ and boost this using the machinery of gradient-boosting (Friedman, Hastie, and Tibshirani 2000; Dietterich, Ashenfelter, and Bulatov 2004). Since our data is naturally relational, we adapt the relational boosted bandits for this case (Kakadiya, Natarajan, and Ravindran 2021).

We have m regression trees for each predicate p, where m is the number of time steps or iterations. Each iteration approximates the corresponding gradient for p, and each of the trees serve as individual components for the final potential function ψ. The algorithm BOOSTEDTREES($predicates$) then loops across all predicates and learns the potentials for each. The set of regression trees for each predicate then forms the structure of the conditional probability distribution and the set of leaves of each tree form the parameters of the conditional distribution.

Algorithm 4 provides the pseudocode for *M3*. We first represent all data in the form of predicates. There are three types of relationships: (1) every call for proposal c_i mapped to a skill set S_i, (2) every researcher r_j mapped to their research interests $\sigma(r_j)$, and (3) every call for proposal c_i teamed with a group of researchers R according to the demand and supply. For each predicate p (shown in BOOSTEDTREES($predicates$)), we generate multiple examples E for our regression-tree learner to get new regression trees and update the current model F_i^p at every iteration i. The function GENERATEEX($p, predicates, F_{i-1}^p$) iterates over all the examples E and computes probability and gradient for each. These probabilities are later used to form teams using a greedy policy, where candidate members with highest probabilities are prioritized first when forming teams.

ULTRA System and Survey Deployment

We built a UI for our system and deployed it using the three use cases detailed in the previous section. ULTRA consists of a three-layered architecture: (1) data storage and retrieval, (2) team matching, and (3) analysis of results. (1) The data we used to perform our experiments is publicly available: (a) calls for proposals from NSF archives, and (b) faculty directories at our university. These are retrieved and stored in a separate database, where they are periodically refreshed to get the latest information. (2) We use the input data and a matching method to view teaming results, along with their respective goodness scores. (3) We evaluate the results both computationally and empirically.

We measure the quality of teaming suggestions, and user satisfaction by conducting a user study for ULTRA over a span of 28 days. We invited researchers from a college within our university to explore the tool and assess its functionality and satisfaction using a feedback survey for every result. The survey includes two 5-point Likert Scale questions: (1) *How relevant is the output given the input?*, and (2) *How useful is the output?*. We also provide a freeform section for additional comments, if any.

Method	Average Quality	Average Volume
M0	0.0879 ± 0.0290	10
M1	0.3673 ± 0.1569	10
M2	0.4097 ± 0.1313	9
M3	0.5295 ± 0.0816	6

Table 1: Average quality (G) and volume of teams ($\#T$) shown *per* researcher (r_j) at USC. This was done for each method M_i, across 434 RFPs and 200 researchers. For average quality, we report the mean and standard deviation, denoted as `mean±STD`.

Evaluation

Computational Evaluation of Output

Quality vs. Volume of Teams. For each method, we assess the quality (goodness) and volume (size) of each teaming suggestion that every researcher r_j receives per every call for proposal c_i. Our experiments iterate across a dataset of 434 RFPs and 200 researchers. For each call, every researcher has a maximum cutoff of 10 teams. For each method, we then find the average goodness (G) of teams (T) a researcher r_j has been recommended. The more advanced the method, the better teams a researcher receives. We observe another unique trade-off as a result, where teams formed algorithmically led to an increased precision and quality, and a notable decrease in quantity. *M0*, random team formation, showed poor quality in results with an average goodness of only 0.0879, despite the number of teams being abundantly available. *M3*, on the other hand, shows a decrease in the number of teaming choices available for a researcher, but a visible increase in quality, compared to the average goodness for *M0*. Table 1 shows the overall results.

Human Evaluation of Output

We deployed our tool at a college-wide level and requested participation from faculty members. This research study has been IRB-approved by our university (IRB# Pro00127449) and reflects an observational (unmonitored) qualitative analysis, where we make the tool publicly available and give participants a demo of its usage but do not actively control their actions. This helps us receive many responses quickly and at a low cost, albeit not without its own limitations. For instance, it does not require users to explore all use cases and methods. Therefore, we only make inferences from data that is recorded and do not draw any from those that are left out. As a possible future work, we can perform a controlled qualitative analysis, where we request participants to come to a designated lab, try every pathway, and give feedback. This will then enable us to answer how a user compared the effectiveness of each method on the same example.

We received a total of 212 responses. Regarding relevancy of outputs, 157 answers were rated as *very relevant* and 34 as *somewhat relevant*, summing to 90.09% of all responses. Similarly, in terms of tool utility satisfaction, 172 answers

	Qualitative Results
Comments	1. *"This is incredible and has a lot of potential. Can't wait for this to be in real time!"*
	2. *"Very well thought out! Great resource to the university"*
	3. *"Lots of new people to choose from here! Great work!"*
	4. *"Very useful tool overall, I could see the practical usage of this work!"*
	5. *"Would love an explanation for all of your methods used!!"*
Feedback	1. *"Seeing many team pitches here from interdisciplinary domains. Can we choose from say, two, settings where we may choose to work with those from a similar domain or a different one? And how so is the overall goodness score calculated?"*
	2. *"A search bar would be great for this one, not just a dropdown!"*
	3. *"In addition to the proposal, can we also add research interest as a user-given input? As a merging of the second and third use cases"*
	4. *"can we build our own teams for a grant?"*

Table 2: A sample of the comments as well as feedback for improvement received from the human study.

responded with *very useful* and 35 with *somewhat useful*, totaling 97.64% of all responses. Figure 2 shows a more quantitative breakdown of the responses and Table 2 highlights a few comments and feedback we received.

We observe two repeated patterns of responses regarding *M0*: (1) Teaming choices that were rated as *'somewhat irrelevant'* or *'irrelevant'*, yet still *'very useful'*, and (2) teaming choices that were rated as *'very relevant'*, despite the irrelevancy in the results. Upon analyzing the comments, some mentioned that *M0* could regardless be useful in working with new colleagues from other departments and *'expand new connections'* as a result. Furthermore, we have only requested users to judge the *usability* and *functionality* of our tool, not immediate applicability. They may also be unaware of each candidate member's skill set, which impedes on their ability to accurately quantify a team's success. Due to that, there is a need and role of AI in teaming, as reflected in comments as well. We also factor this intuition into the interpretation of results.

Discussion - *Ultra as a Deployed Application and Its Generality*

In this section, we discuss the characteristics of ULTRA as required for a deployment track paper but not discussed elsewhere. We describe its development experience leading to deployment at a US university. We also demonstrate its generality by considering an altogether different setting from India but leading to similar results: with increased sophistication of methods, the quality of teams increase and the amount of recommended teams decrease.

Method	Average Quality	Average Volume
M0	0.0896 ± 0.0006	10
M1	0.4218 ± 0.0011	8
M2	0.4292 ± 0.0017	7
M3	0.5835 ± 0.0203	1

Table 3: Average quality (G) and volume of teams ($\#T$) shown *per* researcher (r_j) at IIT-R. This was done for each method M_i, across 100 RFPs and 46 researchers. For average quality, we report the mean and standard deviation, denoted as $\texttt{mean}\pm\texttt{STD}$.

Development and Deployment

The development of ULTRA first began in Summer 2021 with a team of 8 developers and an initial prototype was created within 3 months with M0 and M1 for user feedback (August 2021). It included a very small dataset of researchers and RFPs, a single method (i.e., a string-based matching algorithm and greedy teaming strategy), no goodness score and instead only a match percentage threshold, and a smaller-scale empirical evaluation with a few experts at a single University that indicated the promise of such a tool. Once the pilot study results seemed promising, ULTRA was refreshed with additional data; re-imagined and re-designed in Fall 2022; improved through iterative review, feedback, and testing; enhanced with *M2* and *M3,* and alpha-tested with users from different departments at the University of South Carolina for a year. The system was deployed on May 1, 2023 and evaluated under an IRB-approved protocol for 28 days, and the results are as reported in the paper. It remains publicly available as of writing this paper (November 2023). During July-August 2023, we evaluated ULTRA with data from another setting in a different country: researchers at Indian Institute of Technology-Roorkee (IIT-R), India and calls from India's funding agencies.

One main challenge during development was access to clean data related to RFPs and relevant researcher profiles. Due to inconsistencies in data formatting, missing values, and irrelevant or out-of-date entries, exploring automated approaches had been unsuccessful, and data cleaning had only been possible through iterative collaborations. After deployment, additional challenges were raised. One challenge was maintaining data integrity across changes to ULTRA's servers and infrastructure, and another was change in researchers over time due to hiring or attrition. It was essential to run frequent tests to enhance user experience, create a working feedback loop, and ensure a scalable architecture with minimal overhead.

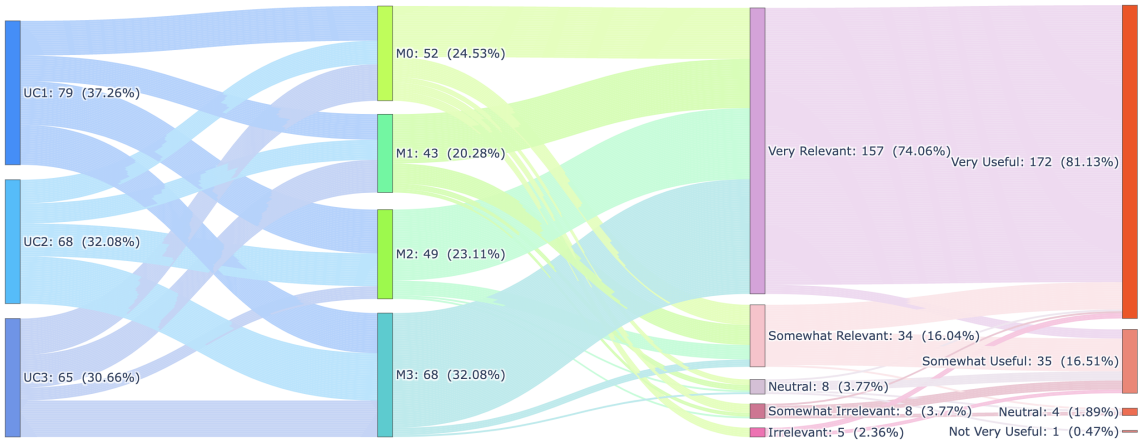

Figure 2: Sankey breakdown of the 212 responses we received from our human study. It shows four categories of nodes: (1) use cases, (2) methods, (3) scale of relevancy, and (4) scale of usefulness. Each line maps a connection from node type to another, showing the flow of interaction that users had with ULTRA.

Generalizing to Second Institution: IIT-R

We additionally extended ULTRA to another university in a different region of the world: Indian Institute of Technology-Roorkee (IIT-R), India. We gathered publicly available data on RFPs from the Department of Science & Technology (DST), a division of Research and Development (R&D) programmes belonging to and funded by the Government of India's Ministry of Science and Technology (DST 2023). We further collected data about IIT-R's faculty members and their respective research profiles. Our initial evaluation is with 100 RFPs and 46 researchers. Table 3 shows the computational evaluation of ULTRA on IIT-R's data. From the quantitative results, we observe a similar trend to Table 1, where teams formed algorithmically led to an increased precision and quality, and visible decrease in quantity. Since human study at any institution is subject to local policies and workplace culture, we leave performance of such a study at IIT-R as a future work.

Conclusion

To conclude, we presented the problem of building teams for funding that allows for collaboration opportunities. We then created and implemented AI methods using string, taxonomy, and more advanced contextual boosted bandits in ULTRA, and demonstrated them to be quite useful both quantitatively (where informed methods increase recommendation quality while reducing their volume) and qualitatively in real human evaluations. We showed the generality of our approach in two different settings from US and India, and discussed our experience of deploying the system.

One area to extend our work is by considering larger data sizes for both researchers and RFPs, and from more diverse sources. Furthermore, since our methods are dependent on open data about researchers as well as proposal calls, this dependency can be both a source of strength and weakness. Data has the potential to encourage teaming without human bias but can also lead to inferior recommendation if the data is obsolete. Similarly, any feedback on the success of recommendation is only possible when a proposal has been won, and this data is usually not available or quite delayed (months or years after recommendation) to be useful. Considering a longer time frame for recommendation with proposal success data could lead to better results. Another area is to further enhance our goodness score with more metrics (e.g., considering the relevance of a researcher's previous projects to current RFPs and the number of ongoing projects a researcher is engaged in).

Our work inspires several interesting future extensions: considering a variety of domain knowledge including but not limited to fairness constraints, teaming constraints, domain constraints, etc. and developing a knowledge-driven learning system that can both exploit both the data and such knowledge remains an exciting future direction. A second direction would be to develop methods that would allow for interactive teaming where the system could not only present the recommendations but explain why and allow for human inputs to be used for refining the learned models. In addition, the scale of our survey could also be improved to ask about more aspects of the teams from participants: diversity and the connection strength between members, etc. A final and important direction is to scale this to different collaborative settings including but not limited to: healthcare, finance, law/legal, mental health, and educational support.

Acknowledgements

We would like to thank Michael Widener, Sugata Gangopadhyay and Sandeep Kumar for their valuable assistance in the data collection and processing for IIT-R; Rohit Sharma and Owen Bond for an earlier version of ULTRA; and Michael Huhns, Danielle McElwain, Michael Matthews, and Paul Ziehl for discussions. Siva Likitha Valluru, Biplav Srivastava, and Sai Teja Paladi acknowledge funding from South Carolina Research Agency, Cisco Research, and VAJRA program, while Siwen Yan and Sriraam Natarajan acknowledge AFOSR under award FA9550-19-1-0391.

References

Abdollahpouri, H.; Burke, R.; and Mobasher, B. 2019. Managing popularity bias in recommender systems with personalized re-ranking. *arXiv preprint arXiv:1901.07555*.

ACM. 2012. ACM Classification Scheme. https://www.acm.org/publications/computing-classification-system/how-to-use. Accessed: 2023-12-01.

Ahmed, F.; Deb, K.; and Jindal, A. 2013. Multi-objective optimization and decision making approaches to cricket team selection. *Applied Soft Computing*, 13(1): 402–414.

Alberola, J. M.; del Val, E.; Sanchez-Anguix, V.; Palomares, A.; and Dolores Teruel, M. 2016. An artificial intelligence tool for heterogeneous team formation in the classroom. *Knowledge-Based Systems*, 101: 1–14.

Amer-Yahia, S.; Roy, S. B.; Chawlat, A.; Das, G.; and Yu, C. 2009. Group Recommendation: Semantics and Efficiency. *Proc. VLDB Endow.*, 2(1): 754–765.

Anagnostopoulos, A.; Becchetti, L.; Castillo, C.; Gionis, A.; and Leonardi, S. 2012. Online team formation in social networks. In *Proc. 21st international conference on WWW*.

Aziz, H.; Brandl, F.; Brandt, F.; Harrenstein, P.; Olsen, M.; and Peters, D. 2017. Fractional Hedonic Games. *TEAC*.

Boratto, L.; Carta, S.; Satta, M.; et al. 2010. Groups Identification and Individual Recommendations in Group Recommendation Algorithms. In *PRSAT@ recsys*, 27–34.

Castro, J.; Quesada, F. J.; Palomares, I.; and Martínez, L. 2015. A consensus-driven group recommender system. *International Journal of Intelligent Systems*, 30(8): 887–906.

Costa, A.; Ramos, F.; Perkusich, M.; Dantas, E.; Dilorenzo, E.; Chagas, F.; Meireles, A.; Albuquerque, D.; Silva, L.; Almeida, H.; et al. 2020. Team formation in software engineering: a systematic mapping study. *IEEE*, 8.

Cremonesi, P.; Koren, Y.; and Turrin, R. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the 4th ACM Conf. on Rec. Sys.*, 39–46.

Dadelo, S.; Turskis, Z.; Zavadskas, E. K.; and Dadeliene, R. 2014. Multi-criteria assessment and ranking system of sport team formation based on objective-measured values of criteria set. *Expert Systems with Applns.*, 41(14): 6106–6113.

Di Penta, M.; Harman, M.; and Antoniol, G. 2011. The use of search-based optimization techniques to schedule and staff software projects: an approach and an empirical study. *Software: Practice and Experience*, 41(5): 495–519.

Dietterich, T. G.; Ashenfelter, A.; and Bulatov, Y. 2004. Training conditional random fields via gradient tree boosting. In *Proc. 21st ICML*, 28.

DST. 2023. Department of Science & Technology. *Government of India — Ministry of Science and Technology*.

Friedman, J.; Hastie, T.; and Tibshirani, R. 2000. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of statistics*, 28(2): 337–407.

Gairing, M.; and Savani, R. 2010. Computing stable outcomes in hedonic games. In *Algorithmic Game Theory: 3rd International Symposium, SAGT 2010*, 174–185. Springer.

Gaston, M. E.; and desJardins, M. 2005. Agent-Organized Networks for Dynamic Team Formation. In *Proc. AAMAS*, 230–237. ACM. ISBN 1595930930.

Gunn, T.; and Anderson, J. 2015. Dynamic heterogeneous team formation for robotic urban search and rescue. *Journal of Computer and System Sciences*, 81(3): 553–567.

He, V. F.; von Krogh, G.; and Sirén, C. 2022. Expertise diversity, informal leadership hierarchy, and team knowledge creation: A study of pharmaceutical research collaborations. *Organization Studies*, 43(6): 907–930.

Juárez, J.; Santos, C. P.; and Brizuela, C. A. 2021. A Comprehensive Review and a Taxonomy Proposal of Team Formation Problems. *ACM Computing Survey*, 54(7).

Kakadiya, A.; Natarajan, S.; and Ravindran, B. 2021. Relational boosted bandits. In *Proceedings of the AAAI Conference on AI*, 13, 12123–12130.

Lazar, M.; Miron-Spektor, E.; Agarwal, R.; Erez, M.; Goldfarb, B.; and Chen, G. 2020. Entrepreneurial team formation. *Academy of Management Annals*, 14(1): 29–59.

Machado, L.; and Stefanidis, K. 2019. Fair team recommendations for multidisciplinary projects. In *IEEE/WIC/ACM International Conference on Web Intelligence*, 293–297.

McLeod, P. L.; Lobel, S. A.; and Cox Jr, T. H. 1996. Ethnic diversity and creativity in small groups. *Small group research*, 27(2): 248–264.

Nawaz, H.; Edmondson, A. C.; Tzeng, T. H.; Saleh, J. K.; Bozic, K. J.; and Saleh, K. J. 2014. Teaming: an approach to the growing complexities in health care: AOA critical issues. *JBJS*, 96(21): e184.

Noll, J.; Beecham, S.; Richardson, I.; and Canna, C. N. 2016. A global teaming model for global software development governance: A case study. In *2016 IEEE 11th ICGSE*.

Okimoto, T.; Schwind, N.; Clement, M.; Ribeiro, T.; Inoue, K.; and Marquis, P. 2015. How to Form a Task-Oriented Robust Team. In *AAMAS*, 395–403.

Roy, S. B.; Thirumuruganathan, S.; Amer-Yahia, S.; Das, G.; and Yu, C. 2014. Exploiting group recommendation functions for flexible preferences. In *2014 IEEE 30th ICDE*.

Srivastava, B.; Koppel, T.; Paladi, S. T.; Valluru, S. L.; Sharma, R.; and Bond, O. 2022. ULTRA: A Data-driven Approach for Recommending Team Formation in Response to Proposal Calls. In *IEEE ICDM Workshops 2022*.

Su, X.; and Khoshgoftaar, T. M. 2009. A survey of collaborative filtering techniques. *Advances in AI*, 2009.

Trope, E. 2023. Why Small Team Collaboration Usually Beats Larger Groups. *Ambition & Balance, undated*.

Valluru, S. L.; Srivastava, B.; and Paladi, S. T. 2023. ULTRA Resources Github. https://github.com/ai4society/ULTRA-Team-Recommendation-Resources. Accessed: 2023-05-31.

Warhuus, J. P.; Günzel-Jensen, F.; Robinson, S.; and Neergaard, H. 2021. Teaming up in entrepreneurship education: does the team formation mode matter? *IJEBR*.

Yalcin, E.; and Bilge, A. 2022. Evaluating unfairness of popularity bias in recommender systems: A comprehensive user-centric analysis. *Info. Proc. & Mgmt.*, 59(6): 103100.