Using Speech Data to Automatically Characterize Team Effectiveness to Optimize Power Distribution in Internet-of-Things Applications

Gnaneswar Villuri
Department of Electrical and Computer Engineering
Stony Brook University
Stony Brook, NY 11794-2350, USA
Email: gnaneswar.villuri@stonybrook.edu

Alex Doboli
Department of Electrical and Computer Engineering
Stony Brook University
Stony Brook, NY 11794-2350, USA
Email: alex.doboli@stonybrook.edu

***Abstract*—This paper focuses on a fresh paradigm where human actions and Machine Learning meet to maximize system performance of Internet-of-Things Edge (IoT-E) with Humans-in-the-Loop applications. To optimally allocate resources, like the available energy, this paper explores the challenges of bridging the semantic gap between team dynamics and team efficiency, so that the more effective teams are given higher priority in resource allocation during operation. The paper proposes methods to interpret team activities using transformer models, like DistilBERT, to process team interactions conducted through speech, and then to utilize the extracted insight to characterize team dynamics. Based on these characteristics, a dynamic power distribution scheme was designed to allocate the available power to teams with higher effectiveness. The results show that the proposed method can improve power allocation in IoT-E applications.**

***Index Terms*—power management, transformer, team behavior, speech processing**

I. Introduction

Internet-of-Things Edge (IoT-E) is a distributed computing paradigm, in which data sensed through a variety of sensors is used to produce diverse actuation responses [1]. There is no server or centralized control in IoT-E. A broad set of performance requirements are specified for an application, like real-time response, robustness, low power/low energy consumption, etc., including the timely requirements for low carbon footprint and decarbonization for sustainability [2]. Moreover, a special case of applications involve the presence of humans acting individually, in teams or in communities as part of IoT-E operation [3]. These applications are often called "Humans-in-the-Loop" (HiL) applications [4]. Finally, Machine Learning (ML) methods [5] are increasingly used in IoT-E together with traditional, algorithmic methods. Combining IoT-E with ML and the requirements of HiL opens up new opportunities in applications, like automotive, smart traffic, smart homes, healthcare, power supply management, and many more.

HiL requirements for IoT-E applications introduce new facets not present in traditional IoT, such as using the extracted insight about human activities (in a privacy-mindful way) to optimize the produced actuation responses and resource management. For example, in smart traffic applications, the cars participating in a traffic situation can arguably form an ad-hoc IoT-E with HiL, the goal being to reduce driving time and fuel use and to maximize the quality of the driving experience [6]. Similarly, IoT with HiL offers the promise to improve patient satisfaction in healthcare, for example by improving the quality of the interactions between patients and healthcare professions [7]. Data on the human activities and behavior can be acquired through sensing devices, like motion detectors, accelerometers, physiological sensors (e.g., perspiration and heart rate sensing), cameras, and microphones. An intriguing situation is that in which the sensed data is the dialog speech during collaborative activities in teams [8]. For example, current voice-processing systems can pick-up certain keywords, like words describing commands for specific actuators, such as "`turn on the light in the kitchen`", but cannot understand more general commands that include degrees of ambiguity, incompleteness, and errors. Instead, voice commands must be unambiguous and deterministic. This is an important limitation in deploying IoT-E for situations that require natural interactions with the participants. New approaches are needed to support understanding natural human speech used as sensed data in IoT-E applications with HiL.

This work focuses on methods to automatically extract facets about the meaning (semantics) of spoken sentences and connect the obtained meaning to a group's behavior, so that the available physical resources, like energy, are distributed to groups depending on how their behavior helps achieve the desired outcomes, like solving a problem. Group behavior is expressed by tracing the verbal interactions of a team by identifying the categories to which sentences belong, i.e. discussing problem requirements, stating high-level solution ideas, elaborating ideas, analyzing ideas, and modifying a solution. Team behavior is then linked to team efficiency to create a predictive model about the expected efficiency. The predictive model is used for power management.

The paper has the following structure. Section II summa-

This work relates to Department of Navy award N00014-22-1-2001 issued by Office of Naval Research.

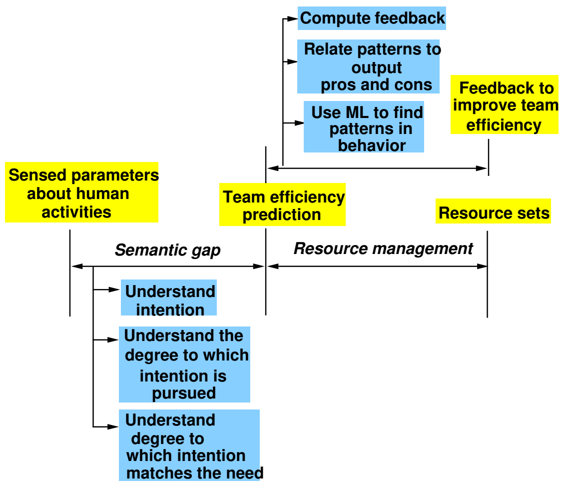

Fig. 1. Data acquisition for IoT-E applications with HiL.

rizes related work and Section III describes the studied problem. Section IV presents the interpretability of the transformer models used in understanding speech. Section V focuses on characterizing team efficiency and Section VI discusses energy distribution using team efficiency. Conclusions end the paper.

II. Related Work

Automatically understanding the meaning of digital information has been a traditional [9] but also a recent topic [10] in ML. Research in computational semantics focuses on identifying the concepts in a corpus, like semantic image segmentation to find the objects in images [11], semantic grouping of design elements [12], and creating dialogues [13] or other outputs similar to human speech [14]. ML methods utilize a variety of statistical methods and metrics computed on the data features, including words, word sets, and word embedding [15]–[17]. ML model interpretation methods are grouped into three categories: structural methods, behavioral methods, and causal methods [18]. Structural methods identify the information learned by the parts of a model. Behavioral models study the degree to which the studied phenomena and processes are captures, e.g., the degree to which a model solves linguistic ambiguities. Causal methods compute the impact of the hidden states of an ML model on the produced output using theories in causal analysis, like causal mediation analysis [10].

This work used Captum method to predict the importance of a feature, neuron, and layer for the final classification [19]. The method computes the importance using gradient and perturbation analysis. This paper suggests a new way to connect the importance assigned to the words in a dialog to the team efficiency, and then uses the found relationships as priority functions to assign the available energy to multiple teams.

III. Problem Description

A. Overview

Fig. 1 summarizes the facets of data acquisition related to the HiL requirements of an IoT-E. Physical parameters are sensed about human activities, e.g., microphones are used to track the human discussions of the participants to a collaborative activity [20]. Such parameters include voice pitch, duration, intensity and quality, signal to noise ratio, and voice activity detection. Also, speech diarization assigns speech segments to the speakers, and speech-to-text conversion produces text transcripts of the spoken data. In this paper, the sensed data are the text representations of spoken dialog.

There is a semantic gap shown in Fig. 1 between the sensed parameters and the prediction of the expected team efficiency in contributing to the problem solution. Team efficiency is the metric used in optimizing the resource management, like allocating the available computing and power resources to teams. Bridging the gap requires three activities as a starting point: (i) Understanding the intention of the team, (ii) Understanding the degree to which the intention is pursued through the specific steps of the team, and (iii) Understanding the degree to which the pursued intention matches the needs of the application, thus producing a contribution of the team to solving the problem. In addition, Fig. 1 shows that the predicted team efficiency is also utilized to create feedback to a team to improve its efficiency. The related tasks are (iv) Using ML methods to identify patterns in the individual and team behavior, (v) Relating the patterns to the outputs of a team and the pros and cons of the outputs (i.e. the output quality), and (vi) Computing the feedback that is given to the team.

Devising automated methods to bridge the semantic gap between sensing and team efficiency prediction, and then using prediction to optimize resource management is the paper topic.

B. Studied application

The following problem was used as a case study for devising semantic gap bridging algorithms, and then using the predicted efficiency as a priority function in energy resource allocation.

Throughout this paper we used two key words: *1. sequence* is a sentence of words or an instance, and *2. category* describes one of the five classes (or color labels) assigned to a sequence.

We conducted an experiment, in which undergraduate students (UGs) were asked to work on a programming exercise as a group of three students for 20 minutes. We recorded their conversations throughout and transcribed them to text using the software tool presented in [20]. We manually labeled each sequence into five categories, each indicating a step of the problem solving process. Then the dynamics of problem solving can be expressed by tracing the categories to which the dialog sentences pertain. The following five categories were considered, each category being labeled with a distinct color to shorten the presentation: (i) discussing problem requirements (color yellow), (ii) analyzing solution ideas (color green), (iii) describing high-level solution ideas (color grey), (iv) elaborating the high-level ideas (color blue), and (v) modifying an existing solution (color red). The dynamics of a team's problem solving process is then linked to the team's efficiency, which was also manually labeled. This linking creates a predictive model that was used to anticipate a team's expected efficiency, and utilized for energy management, like defining a scheme to give more energy to the more efficient teams.

Thirty teams participated to the conducted experiment, and a separate dataset was collected for each team. The dataset sizes were between 56 and 289 sentences, with a total of 3714 sentences. Six datasets were used in our experiments.

IV. Interpretability of Transformer Models

DistilBERT transformer [21] was trained to predict the category of each input spoken by a participant to a team. Moreover, the interpretability information for the transformer classification was needed, so that the main word structures that decided a category could be identified and the team dynamics characterization related to features, like the degree to which ideas were repeated, corrected, detailed, matched the problem requirements, or addressed mistakes. These features were then linked to the manual characterization of a team's activity to create the prediction model used in energy management.

A. Overview

Transformer models are proving to be one of the best current modeling alternatives for any given Natural Language Processing (NLP) task. This is because of their ability to capture long-range dependencies, unlike Recurrent Neural Networks (RNNs), which operate on sequential data resulting in vanishing gradients, hence, making it difficult for the model to learn long-range dependencies [22]. The self-attention and parallel processing mechanisms of transformer models make them efficient in tasks like these. Self-attention is the process of assigning weights to each word of a sequence based on its importance with respect to every other word [21].

B. Interpertability

The performance of advanced Neural Networks, like RNN [22], Long-short term memory (LSTM) [22], or transformers comes at a cost. The cost is their interpretability. These so-called "black-box" models are arguably not as interpretable as other classification models, like Decision Trees [23], because of their convoluted architecture. However, recent attention visualization techniques, like transformers-interpret [24], have made interpretability possible.

We used three techniques to interpret the created transformer models for sentence classification into the five categories:

1) *Attribution Score*: The metric is the score assigned by the model to each word of a sequence based on its importance or influence in making the final prediction. The words with high attribution scores are the most influential and effective while the ones with low scores are the least effective on the final prediction. Positive attribution scores mean that the words positively influenced the predicted category and negative attribution scores mean that they negatively influenced the predicted category. Fig. 2 shows the attribution scores of the sequence "`I don't know how to do that.`" against LABEL_0 (blue category).
2) *Visualizing Class Attributions*. When the sequences get long, using numbers like Attribution Scores might become arduous to read. Visualizing those numbers makes interpretability effortless. We utilized the function called `visualize()` of transformer-interpret [24]. The function creates an HTML file by highlighting the attributions against each category by utilizing Captums's inbuilt visualization library [24]. Fig. 3 shows the visualization, where LABEL_0 indicates the blue category, LABEL_1 is for the green category, LABEL_2 refers to the grey category, LABEL_3 indicates the red category, and LABEL_4 designates the yellow category.
3) *Word Clouds*. This is another visualization technique that helps identify the most effective words for each category. Fig. 4 shows the word cloud of positive words for category green. The word size indicates the word importance for assigning a sentence to a specific category.

```
[CLS]: 0.0
i: 0.09783403486919981
don: -0.36781506456827096
': -0.21559877892392995
t: -0.3491895214898905
know: -0.3467651964990422
how: -0.5120951438343142
to: -0.3827236506178983
do: -0.29832481241928677
that: -0.25472965095030636
.: -0.06224632634748551
[SEP]: 0.0
```

Fig. 2. Examples of Attribution Scores assigned to the words and punctuation marks of a sentence.

Legend: Negative Neutral Positive

n/a	Prediction Score	Attribution Label	Attribution Score	Word Importance
n/a	(0.85)	LABEL_0	-0.08	[CLS] we ' ll have 20 minutes to work on it . [SEP]
n/a	(0.88)	LABEL_1	2.51	[CLS] we ' ll have 20 minutes to work on it . [SEP]
n/a	(0.21)	LABEL_2	-2.07	[CLS] we ' ll have 20 minutes to work on it . [SEP]
n/a	(0.09)	LABEL_3	-2.06	[CLS] we ' ll have 20 minutes to work on it . [SEP]
n/a	(0.41)	LABEL_4	0.25	[CLS] we ' ll have 20 minutes to work on it . [SEP]

Fig. 3. Visualized Attribution Scores.

V. Characterization of the Team Activities

In the process of characterizing the team dynamics during problem-solving, we defined six characteristics of the solving steps and the interactions in a team. These characteristics describe the different facets of team collaboration, arguably

Fig. 4. Word Cloud of positive words with a greater influence in classifying a sentence as category green (analyzing solution ideas).

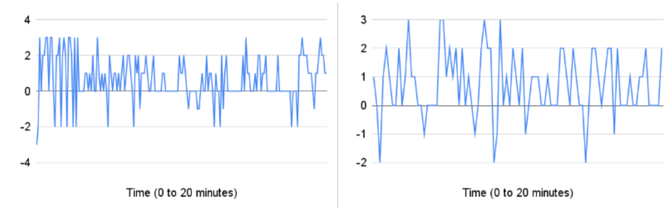

Fig. 5. Discussions correctness of a more effective team (left figure) and a less effective team (right figure).

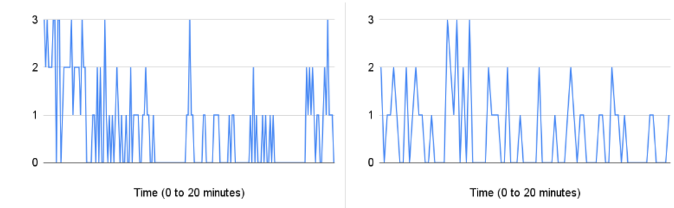

Fig. 6. Discussions match with problem requirements of a more effective team (left figure) and a less effective team (right figure).

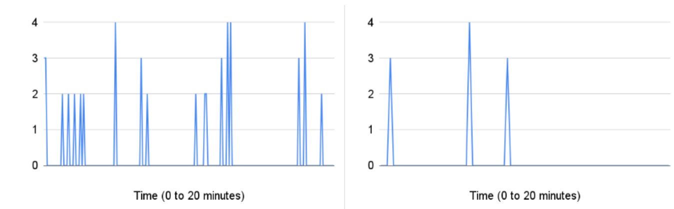

Fig. 7. Degree of correcting mistakes of a more effective team (left figure) and a less effective team (right figure).

allowing to algorithmically distinguish (e.g., using an ML method) between the more and less effective teams.

1) *Repetitiveness*. Repetitiveness indicates how similar the current sequence is with respect to the previous sequences. High repetitiveness suggest little or no progress in solving the problem as discussions are fixated on a set of ideas. Low repetitiveness suggests progress, if there is connection with the previous sentences, like shared words. If there is little connection then low repetitiveness suggests that discussions moved on to a different idea.
2) *Uniqueness*. Uniqueness captures the inclusion of new ideas and perspectives within the solution by a team. Teams with high uniqueness indicate active exploration of solutions when new alternatives are identified and analyzed. Teams with low uniqueness have minor or no contribution in attaining the solution.
3) *Correctness*. Correctness measures the accuracy of the discussion in terms of programming context or problem solving in general. Positive correctness suggests that the team is likely moving in the right direction towards solving the problem. Negative correctness indicates mistakes in their idea, reasoning, analysis, or implementation.
4) *Matching with problem requirements*. This characteristics assesses how well the team's communication aligns with the requirements of a programming exercise, e.g., problem description. High alignment indicates a clear understanding of the problem. Low alignment suggests unclear or incorrect understanding of the problem needs.
5) *Degree of correcting mistakes*. The characteristic describes the ability of the team members to identify and rectify errors or misconceptions in their reasoning, analysis, and implementation. Teams with a high degree of correcting mistakes demonstrate strong problem-solving skills and collaborative learning, as errors are likely during problem solving. Conversely, low correction rates indicate a reluctance to challenge potential issues.

The paper argues that the six characteristics can be used to distinguish between teams depending on their effectiveness, like low efficiency teams, average efficiency teams, and high efficiency teams. For example, as shown in Fig. 5, an effective team demonstrated a better frequency of accurate discussions (value >=3 on a scale of -4 to 4) 15 times over a time period of 20 minutes, whereas the least effective team only managed 5. This indicates a disparity between the high efficiency and low efficiency teams in terms of their capabilities of having accurate discussions. Similarly, Fig. 6 shows the discussions matching with the problem requirements. The average value of a more effective team is 0.7 whereas it is 0.6 for the less effective team. This indicates that the more effective team had discussions more closely to the problem requirements in contrast to the less effective team. Finally, Fig. 7 presents the degree of correcting mistakes. The more effective team corrected their mistakes 17 times whereas the less effective corrected their mistakes just 3 times, showing that the more effective team was superior at finding and addressing errors, thus further demonstrating problem-solving skills. However, more experiments are needed to strengthen these observations.

TABLE I
POWER AVAILABILITY SCENARIOS.

Time Interval	Scenario 1	Scenario 2	Scenario 3	Scenario 4
0 - 5	0	2400	2900	2600
5 - 10	1000	3100	3100	2800
10 - 15	3400	3000	1500	1000
15 - 20	1500	2700	3050	1500
20 - 25	0	2300	3090	3000

TABLE II
PRIORITIES OF TEAMS BY METRICS

Metric	T1	T2	T3	T4	T5	T6	Correlation(%)
Sum	**1**	**2**	**5**	**3**	**4**	**6**	**84**
Mean	2	4	3	1	6	5	58
Median	2	4	3	1	6	5	58
Max	1	2	3	4	6	5	76
Min	4	3	5	2	6	1	17

VI. DYNAMIC POWER DISTRIBUTION BASED ON TEAM EFFECTIVENESS

We investigated electric power distribution across teams with varying degrees of effectiveness, e.g., good, average,

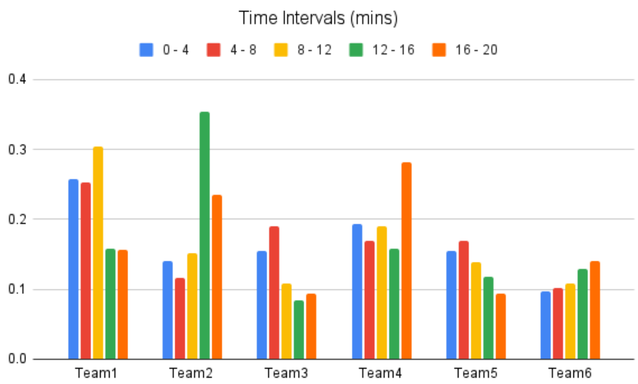

Fig. 8. Normalized priorities of teams by considering sum as a metric.

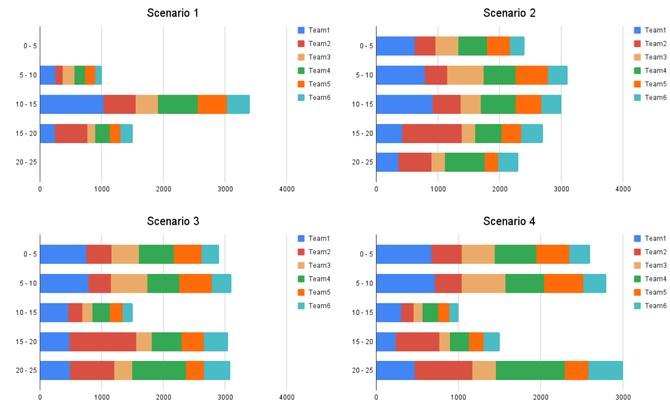

Fig. 9. Power distribution to teams across 4 scenarios.

and bad. Our goal was to dynamically allocate the power to the teams based on their effectiveness. Four different power availability scenarios were considered in the study, as shown in Table I. The six characteristics mentioned in Section V gave us some insight on the team behavior and effectiveness. Among the six, we picked three uncorrelated characteristics (i.e. correctness, matching with problem requirements, and degree of correcting mistakes), and the manually-labeled team efficiencies were related to the three characteristics. The effectiveness of a team was modeled using the following steps:

1) The entire conversation of each team was divided into five time intervals (each interval was 4 minutes long) to reduce the analysis granularity, so that the team behavior was approximately constant over an interval.
2) One of the five metrics (sum, mean, median, max, and min) was utilized to aggregate the values of each characteristic. This step offered different ways of characterizing a team's behavior.
3) A priority was assigned to each team over a time interval for each characteristic from 1 to 6 based on their aggregated value.
4) The mean of the three priorities was computed across three characteristics.
5) The priorities were inverted and normalized, so that the team with the highest priority receives the most power.
6) The power was distributed across teams based on their normalized priorities.

Six teams (labeled as T1 to T6) and their assigned priorities from 1 to 6 with respect to each metric are shown in Table II. We wanted the normalized priorities that were calculated to match the actual effectiveness of the teams. For this reason, we added the column called "correlation" that checks the correlation between the calculated priorities and the actual team performance manually rated by human raters.

Figure 8 depicts the normalized priorities of the six teams computed separately for each of the five time intervals into which the experiment time was divided.

A. Discussion of the Results

The results of the correlation analysis across metrics (sum, mean, median, max, and min) indicated that the metrics with the highest correlation percentages, both above 75%, were the sum and max (in Table. II). The metrics mean and median gave exactly the same correlation values and the metric min was the lowest of all.

- *Min metric has the lowest correlation value.* Even the more effective teams committed some mistakes, had discussions that didn't match the problem requirements, and failed to correct their mistakes. When considering the metric min, these worst cases got picked up, since the minimum values of the characteristics of good teams and bad teams were not different, hence, they became almost indistinguishable.
- *Mean and median have the same correlation values.* The values of the metrics mean and median for any given time intervals were very similar. They differed by only 6% on average. This difference was not significant enough to change the priorities assigned to the teams, as the correlation values were exactly the same.
- *Max has the second highest correlation value.* The reason for the high correlation of the max metric was that for any given time interval, the maximum value of any characteristic of a more effective team was always either more than or equal to the less effective team. Hence, it resulted in priorities that matched with the actual performance, further showing a higher correlation value.
- *Sum has the highest correlation value.* The most important factors that differentiated a more effective and less effective team were the consistency and accuracy in their communication. If a team interacted more frequently, then the number of sequences in a given time interval was higher, so the sum was also be higher. This resulted in priorities that exactly matched their actual performance, further resulting in the highest correlation value.

The normalized priorities of the teams across different time intervals, when the metric sum was chosen, are shown in Table. I. Teams 1 and 2, which are the best teams in terms of

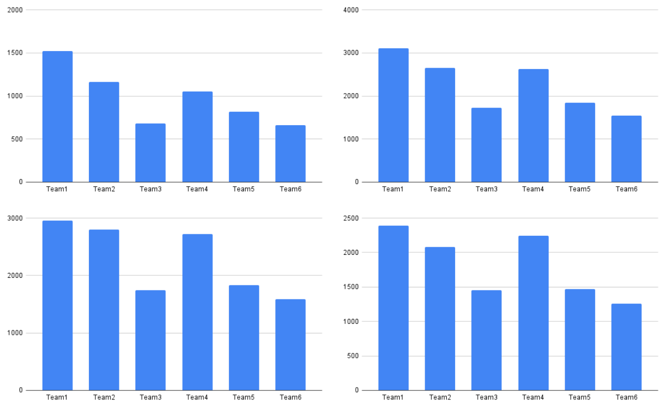

Fig. 10. Total power usage of teams across 4 scenarios.

performance, were assigned the highest priorities. Surprisingly, Team 3, which is an average team, got a lower priority than Teams 5 and 6, which are bad teams. This is because of the poor interaction within the team. Though they were successful in getting the expected output, the total sequences of the group were just 69 (the lowest among all), hence indicating poor interaction. Because the metric sum was considered, the count of the sequences influenced the priority assigned to the team.

The power distribution to the six teams and the total power assignment per team using the metric sum are shown for four power availability scenarios in Figures 9 and 10, respectively. By looking at the power distributions across the five intervals, it can be seen that teams 1 and 2, which are the most effective teams, got the highest proportion of power. However, the power distribution is not linear over the entire experiment time. Team 1 started strong and solved most of their problem in the first 12 minutes of the experiment, but they did not do much later. Team 2, which is another good team, started slow but did progress a lot in the last 8 minutes. Fig. 9 shows that Team 1 got the highest proportion of power in the first three intervals but Team 2 got the highest proportion in the last two intervals.

The overall data indicates that the goal of distributing the power according to the team's effectiveness was achieved through the proposed prioritization technique based on the team's effectiveness modeled using the extracted insight about the meaning (semantics) of the team discussions.

VII. Conclusion

This paper explains the significance of bridging the semantic gap between the verbal interactions in teams, team behavior, and resource allocation in IoT-E with HiL applications. The paper proposes a method to automatically understand the patterns in team behavior to enhance resource management, like energy, within IoT-E frameworks. The proposed characterization of the team behavior and the related dynamic power distribution technique provide possibilities for resource allocation optimization in applications involving collaborative problem-solving processes. Hence, this work pushes the boundaries of IoT-E with HiL applications and also contributes to extending the paradigm of human-centered computing.

Future work will focus on experimentally verifying the proposed resource management strategy, e.g., if assigning more power to teams using their predicted effectiveness is indeed producing better final outcomes. Another opportunity is to validate the proposed approach for other HiL applications with different characteristics, i.e. different problem types [25].

References

[1] J. Hong et al., "Internet of Things (IoT) Edge Challenges and Functions", *Internet Research Task Force (IRTF)*, RFC: 556, April 2024, ISSN: 2070-1721.

[2] "Infineon Drives Decarbonization and Digitalization for a Greener Future with Innovative Semiconductor Solutions at PCIM Europe 2023", *Infineon, Press Release, 19 April 2023*, https://www.infineon.com/cms/en/about-infineon/press/market-news/2023/INFXX202304-091.html.

[3] A. Doboli et al, "A novel agent-based, evolutionary model for expressing the dynamics of creative open-problem solving in small groups", *Applied Intelligence*, Vol. 51, pp.2094-2127, 2021.

[4] X. Wu and et al., "A Survey of Human-in-the-loop for Machine Learning", *Future Generation Computer Systems*, Vol.135, 2022, pp.364-381.

[5] J. Han et al., "Data Mining. Concepts and Techniques", *Elsevier*, 2012.

[6] F. Arena et al., "An Overview on the Current Status and Future Perspectives of Smart Cars", *Infrastructures*, Vol. 5, no. 7: 53. 2020.

[7] A. Stefanini et al., "Patient Satisfaction in Emergency Department: Unveiling Complex Interactions by Wearable Sensors", *Journ.Busin.Res.*, vol.129, pp.600-611, 2021.

[8] R. Duke and A. Doboli, "Applications of diaLogic System in Individual and Team-based Problem Solving Applications", *IEEE International Symposium on Smart Electronic Systems*, 2022.

[9] D. Lenat and G. Marcus, "Getting from Generative AI to Trustworthy AI: What LLMs might learn from Cyc", *arXiv*:2308.04445, 2023.

[10] K. Meng, D. Bau, A. Andonian and Yonatan Belinkov, "Locating and Editing Factual Associations in GPT", *arXiv*:2202.05262, 2023.

[11] Z. Yang et al., "Fully Convolutional Network-Based Self-Supervised Learning for Semantic Segmentation", *IEEE Trans. Neural Networks*, Vol.35, no. 1, pp. 132-142, 2024.

[12] P. Duan et al., "Towards Semantically-Aware UI Design Tools: Design, Implementation, and Evaluation of Semantic Grouping Guidelines", *Proc. Workshop AI and HCI*, 2023.

[13] R. Thoppilan et al., "LaMDA: Language Models for Dialog Applications", *arXiv*:2201.08239, 2022.

[14] A. Srivastava et al., "Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models", *arXiv*:2206.04615, 2023.

[15] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *arXiv*:1810.04805, 2018.

[16] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach", *arXiv*:2019, 2019.

[17] V. Sanh et al., "DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter", *arXiv*:2020, 2020.

[18] J. Vig et al., "Causal Mediation Analysis for Interpreting Neural NLP: The Case of Gender Bias", *arXiv*:2004.12265, 2020.

[19] N. Kokhlikyan et al., "Captum: A unified and generic model interpretability library for PyTorch", *arXiv*:2009.07896, 2020.

[20] R. Duke and A. Doboli, "diaLogic: Non-Invasive Speaker-Focused Data Acquisition for Team Behavior Modeling", *arXiv*:2209.00619, 2022.

[21] A. Vaswani et al., "Attention Is All You Need", *arXiv*:1706.03762, 2023.

[22] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network", *Physica D: Nonlinear Phenomena*, vol. 404, pp. 132306, Mar. 2020.

[23] J. Lin et al., "Generalized and Scalable Optimal Sparse Decision Trees", *Proc. International Conference on Machine Learning*, vol. 571, 2020.

[24] C. Pierse, "Transformers Interpret", version 0.5.2, 2021.

[25] A.Doboli et al., "Modeling Semantic Knowledge Structures for Creative Problem Solving: Studies on Expressing Concepts, Categories, Associations, Goals and Context", *Know.-based Systems*, 78, pp.34–50, 2015.