International Journal of Semantic Computing

Learning a Generalizable Model of Team Conflict from Multiparty Dialogues

Ayesha Enayet
Department of Computer Science, University of Central Florida
Orlando, 32816, US
ayeshaenayet@knights.ucf.edu

Gita Sukthankar
Department of Computer Science, University of Central Florida
Orlando, 32816, US
gitars@eecs.ucf.edu

Received (15 Apr 2021)
Revised (NA)
Accepted (NA)

Good communication is indubitably the foundation of effective teamwork. Over time teams develop their own communication styles and often exhibit entrainment, a conversational phenomena in which humans synchronize their linguistic choices. Conversely, teams may experience conflict due to either personal incompatibility or differing viewpoints. We tackle the problem of predicting team conflict from embeddings learned from multiparty dialogues such that teams with similar post-task conflict scores lie close to one another in vector space. Embeddings were extracted from three types of features: 1) dialogue acts 2) sentiment polarity 3) syntactic entrainment. Machine learning models often suffer domain shift; one advantage of encoding the semantic features is their adaptability across multiple domains. To provide intuition on the generalizability of different embeddings to other goal-oriented teamwork dialogues, we test the effectiveness of learned models trained on the Teams corpus on two other datasets. Unlike syntactic entrainment, both dialogue act and sentiment embeddings are effective for identifying team conflict. Our results show that dialogue act based embeddings have the potential to generalize better than sentiment and entrainment based embeddings. These findings have potential ramifications for the development of conversational agents that facilitate teaming.

Keywords: teamwork; process conflict; multiparty dialogues; entrainment; sentiment analysis; dialogue acts; embeddings; generalizability

1. Introduction

The aim of our research is to create agents who can assist human teams by intervening when teamwork goes awry. To do this, it is important to be able to rapidly assess the status of team performance through "thin-slicing", making accurate classifications from short behavior samples; Jung suggests that developing

this capability would remove the need for developing continuous team monitoring systems [1]. Ambady and Rosenthal demonstrate that many types of social interactions remain sufficiently stable that even a small sample is meaningful at predicting long term outcomes, the most famous application of this theory being thin-slicing marital interactions to predict divorce outcomes [2, 3].

Conflict in teams can be classified as being relationship or task-oriented [4]. *Relationship conflict* arises from "interpersonal incompatibility among members, which typically includes tension, animosity, and annoyance among members within a group" [5]. Our work centers on *task conflict*, "disagreement among group members about the content of the tasks being performed, including differences in viewpoints, ideas, and opinions" [5]. Rather than developing specific measures for predicting future team conflict, we demonstrate that an embedding grouping teams with similar conflict levels can be learned directly from multiparty dialogue. An advantage is that this approach avoids the necessity of collecting advance data on team members, such as personality traits or training records.

This article compares the performance of three types of embeddings extracted from: 1) dialogue acts, 2) sentiment polarity, and 3) syntactic entrainment; these features were selected based on previous work on team communications and group problem-solving. Dialogue acts capture the interactive pattern between speakers in multiparty communication [6]. During dialogue act classification, utterances are grouped according to their communication purpose.

Sentiment polarity measures the attitude or emotion of the speaker during conversation; it can be used to detect disagreement. Entrainment is the natural tendency of the speakers to adopt a similar style during a conversation, causing them to achieve linguistic alignment. There are several types of entrainment including lexical choice [7], style [8], pronunciation [9], and many others [10]. Reitter and Moore demonstrated that syntactic entrainment, based on alignment of lexical categories, can be used to predict success in task-oriented dialogues [7].

Good team communication exhibits all these characteristics: greater emphasis on problem solving than arguing, positive sentiment, and communication synchronization [11]. Our research was primarily conducted on the Teams corpus [12] which consists of player dialogue during a cooperative game. One advantage of studying a clearly defined, time-bounded team task is that the dialogues can be divided into teamwork phases: 1) early (knowledge building) 2) middle (problem solving) and 3) late (culmination). For thin-slicing, we seek to predict the team performance from the initial teamwork stages. The Teams corpus includes team conflict scores, which measure the amount of disagreement that occurred during gameplay. Our hypotheses are:

H1: an embedding leveraging dialogue acts will be useful for classifying team performance at all phases since it directly detects utterances related to conflict (eristic dialogues).

H2: sentiment analysis will consistently reveal team conflict and thus be a good

predictor of performance.

H3: the entrainment embedding will be predictive when the entire dialogue is considered, but will be less useful at analyzing early phases before entrainment has been established.

Embeddings are mechanisms for mapping high-dimensional spaces to low-dimensions while only retaining the most effective representations, making it possible to apply machine learning on large inputs by representing them in the form of sparse vector. Unfortunately, there is a paucity of high quality data on team communications. Thus it is beneficial to learn *generalizable* embeddings that are applicable across multiple datasets. We hypothesize that:

H4: Embeddings based on sequences of dialogues acts will generalize well at predicting task conflict across datasets.

We believe that teams who frequently engage in arguments have very different dialogue act sequences than teams who agree on the future course of action. This article presents our approach for extracting generalizable embeddings from multiparty dialogues that encode team conflict. The next section describes the rich literature on analyzing team communication and multiparty dialogues.

2. Related Work

Team communication, both spoken or written, is a critical element of collaborative tasks and can be studied in a variety of ways. Semantic analysis centers on the meaning of utterances, while pragmatics involves identifying speech acts [13]; both analytic approaches are important and often occur in parallel. In many studies of team communication, this analysis is arduously done through hand coding the utterances.

Parsons et al. [14] contrast two different schemes to code utterances in team dialogues as part of their long term research goal of developing a virtual assistant for human teams. Their comparison illustrates the benefits and problems of the Walton and Krabbe typology [15], which includes categories for information-seeking, inquiry, negotiation, persuasion, deliberation, and eristic, but does not consider the context in which the utterance occurs. The McGrath theory of group behavior [16] focuses on modes of operation: inception, problem-solving, conflict resolution, and execution. When applying the McGrath theory of group behavior, utterance classification is modified by conversational context.

Sukthankar et al. also used an explicit team utterance coding scheme towards the problem of agent aiding of ad hoc, decentralized human teams to improve team performance on time-stressed group tasks [17]. Unlike teamwork studies, we do not specifically map individual utterances to team communication categories, but leverage dialogue act classification models to identify features that are indicative of team conflict. Shibani et al. [18] discussed some of the practical challenges in designing an automated assessment system to provide students feedback on their

teamwork competency: 1) dialogue pre-processing, 2) assessing teamwork chat text, and 3) classifying teamwork dimensions. They evaluated the performance of rule-based systems vs. supervised machine learning (SVM) at classifying coordination, mutual performance monitoring, team decision making, constructive conflict, team emotional support, and team commitment. Even with dataset imbalance, the SVM model generally outperformed the hand coded rules. Our proposed method can also be used to assist human teams by proactively warning them of deficiencies during the early phases of team tasks, without the onerous data labeling requirements.

Other analytic techniques focus on linguistic coordination between speakers in groups. For instance, Danescu et al. studied the effect of power differences on lexical category choices during goal-oriented discussion [19]. This is one form of entrainment in which the speakers preferentially select function-word classes used by other group members. Our article uses a dataset (Teams corpus), that was created to study entrainment in teams [12]. Rahimi and Litman demonstrated a method for learning an entrainment embedding to predict team performance [20]; we use a modified version of their technique to express syntactic entrainment. However since entrainment develops over time, we compare the performance of entrainment at early vs. late task phases. Furthermore, they only focused on syntactic/lexical features of utterances, not semantic.

Sentiment analysis has been applied to the study of group dynamics; for instance, researchers have leveraged sentiment features to detect communities in social networks [21, 22]. Our work demonstrates the utility of sentiment features towards predicting team conflict and show that the sentiment-based embedding is useful during all teamwork phases. We rely exclusively on the multiparty team dialogues; however there have been many attempts to predict team performance using other types of multimodal features. TCdata, a team cooperation dataset, includes both audio and video recordings of teams performing cooperative tasks [23]. Liu et al. explicitly extracted 159 features from team speaking cues, individual speaking time statistics, and face-to-face interaction cues to predict team performance on this dataset.

Several studies [24, 25] have shown team member personality traits to be useful predictors of conflict and team performance. Yang et al. used individual personality traits to predict the performance of final year student project teams using neural networks [24]. Omar et al. developed a student performance prediction model that included both personality types and team personality diversity [25]. Even though these additional data sources can be highly predictive, they are rarely available in real-world team scenarios, unlike multi-party dialogue which is often self-archived to preserve organizational memory.

Marlow et al. [26] conducted a meta-analysis combining data from multiple studies on how team communication relates to team performance. They confirm that communication is positively related to team performance, but that the quality of communication is more important than frequency. This indicates that fre-

quency of utterances alone, lacking information about dialogue acts, sentiment, or entrainment, is unlikely to be predictive of team performance. According to their meta-analysis, task type moderates the relationship between communication and performance; the relationship is strong when team tasks are cognitive (vs. action-based) and have interdependencies. All of our datasets describe tasks that fall into that category.

3. Problem Statement

We aim to design proactive assistant agents that can promote effective teamwork by providing timely assistance. This requires a way to predict when intervention is required. This article makes three research contributions towards this overarching goal.

Encoding team communication with embeddings: This study compares different methods of predicting team conflict. The first approach is to generate embeddings from sequential utterance patterns. In our experiments, the multiparty dialogue is converted either to a sequence of dialogue acts or sentiments which is then used to generate the embedding. These embeddings represent meaningful information about how the communication between the team members is evolving. The second approach is to create an embedding that encodes entrainment relationships between team members. To do this, we map the whole multiparty dialogue to a feature vector representing entrainment in the teams by employing the method proposed by Rahimi et al. [20].

Conflict prediction during initial teamwork phases: During task completion, teams pass through different cognitive phases, starting from brainstorming and completing with problem solution. We compare the performance of different embeddings over teamwork phases: 1) knowledge discovery 2) problem solving and 3) culmination. We show that the sequential embeddings (dialogue act and sentiment) perform well at predicting conflict even during early teamwork phases.

Generalizability across datasets: Supervised machine learning models trained on one dataset, often do not perform well on unseen datasets; this phenomenon is called domain shift [27, 28, 29]. We test models learned on the Teams corpus on datasets gathered from software engineers (GitHub issue comments) and military teams to provide intuition on the generalizability of the embeddings on unseen datasets.

4. Method

This section describes our procedure for computing embeddings using doc2vec [30], an unsupervised method that is used to create a vector representation of the team dialogue. We compare the performance of different possible inputs to doc2vec: 1) dialogue acts, 2) sentiment analysis, and 3) syntactic entrainment.

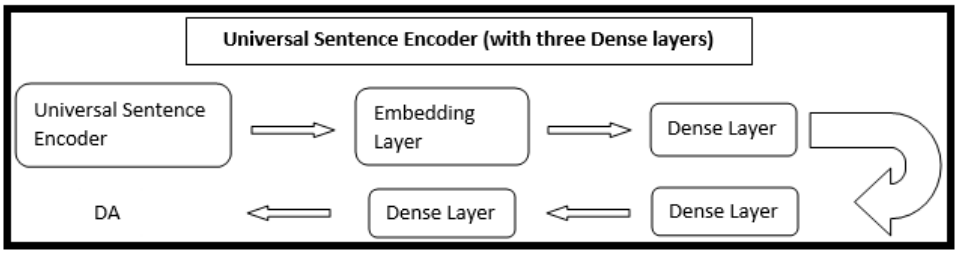

Fig. 1: Dialogue Act Classifier Architecture.

Table 1: Dataset Statistics

Dataset	#Utterances	#Tokens
SwDA	200,052	19,000
Teams Corpus	110,206	573,200

Table 2: SwDA Dataset Sample

Speaker	Utterance	DA	Description
A	I don't, I don't have any kids.	sd	Statement-non-Opinion
A	I, uh, my sister has a, she just had a baby,	sd	Statement-non-Opinion
A	he's about five months old	sd	Statement-non-Opinion
A	and she was worrying about going back to work and what she was going to do with him and –	sd	Statement-non-Opinion
A	Uh-huh.	b	Acknowledge
A	do you have kids?	qy	Yes-No-Question
B	I have three.	na	Affirmative non-yes Answer
A	Oh, really?	bh	Backchannel in question form

4.1. *Dialogue Acts*

Dialogue acts can be created from the semantic classification of dialogue at the utterance level to identify the intent of the speaker. A transfer learning approach was used to tag utterances of the Teams corpus using the DAMSL (Discourse Annotation and Markup System of Labeling) tagset. Figure 1 shows the architecture of our

Table 3: Teams Dataset Sample

Speaker	Utterance	DA	Description
A	Ok I'm going to	sd	Statement-non-Opinion
A	shore up these two.	sd	Statement-non-Opinion
B	Good move.	ba	Appreciation
A	Then we got one and then I guess I can also	sd	Statement-non-Opinion
A	Can I use my powers twice in one play	sd	Statement-non-Opinion
C	Mm	b	Acknowledge (Backchannel)
B	yes	ny	Yes answer

dialogue act classifier, which was constructed using the Universal Sentence Encoder; we selected USE for its ability to achieve consistently good performance across multiple NLP tasks [31]. There are two different variants of the model: 1) a transformer architecture, which exhibits high accuracy at the cost of increased resource consumption and 2) a deep averaging network that requires few resources and makes small compromises for efficiency. The former uses attention-based, context-aware encoding subgraphs of the transfer architecture. The model outputs a 512-dimensional vector. The deep averaging network works by averaging words and bigram embeddings to use as an input to a deep neural network. The models are trained on web news, Wikipedia, web question-answer pages, discussion forums, and the Stanford Natural Language Inference (SNLI) corpus, and are freely available on TF Hub.

We selected the USE Transformer-based Architecture model with three dense layers and a softmax activation function. Figure 1 shows the architecture of our DA classification model, which achieves a validation accuracy of 70%.

The model was fine-tuned using the Switchboard Dialogue Act Corpus (SwDA) dataset. SwDA is one of the most popular public datasets for DA classification. It consists of 1155 human-to-human telephone speech conversations, tagged using 42 tags from the DAMSL tagset. Table 1 shows the statistics of both SwDA and the Teams corpus.

Table 2 shows examples from the SwDA training dataset, and Table 3 shows examples from Teams corpus. Each team dialogue generates a unique sequence where each element of the sequence represents the dialogue act of the corresponding utterance. This sequence of dialogue acts is then used as an input to doc2vec algorithm to create the embedding.

4.2. *Sentiment Analysis*

Another option is to represent the team dialogue as a series of changes in the emotional state of the team. This can be done by applying sentiment analysis to the individual utterances. Sentiment analysis is the task of predicting the emotion or attitude of the speaker; we are using the TextBlob python implementation [32] to determine sentiment polarity of each utterance in the dialogue. The polarities are float values which lies between -1 and 1 representing negative, positive and neutral sentiment. For each team the unique sequence of these polarities is used as input to doc2vec, where each element of the sequence represents the polarity of the corresponding utterance. This representation encodes transitions in the emotional state of the team across the duration of the task.

4.3. *Entrainment*

Entrainment is one form of linguistic coordination in which team members adopt similar speaking styles during conversation. Here we evaluate the performance of a syntactic entrainment embedding based on Rahmi and Litman's [20]'s work that encodes the propensity of subsequent speakers to make similar lexical choices. Eight

lexical categories were used: noun (NN), adjective (JJ), verb (VB), adverb (RB), coordinating conjunction (CC), cardinal digit (CD), preposition/subordinating conjunction (IN), and personal pronoun (PRP) . To calculate the entrainment between two speakers we follow the method proposed by Danescu et al. [19] shown in Equation 1. $Ent_c(x, y)$ is the entrainment of speaker y to speaker x, c is the lexical category, e_{yx^c} represents the event where speaker y utterance immediately follows the speaker x utterance and contains c, e_x^c is the event when utterance (spoken to y) of speaker x contains c.

$$Ent_c(x, y) = p(\frac{e_{yx^c}}{e_x^c}) - p(e_{yx^c}) \tag{1}$$

The NLTK part-of-speech (POS) tagger was used to tag all the utterances with their respective lexical categories. A directed weighted graph was generated for each dialogue linking speakers with positive entrainment. The structure of this graph encodes the entrainment relationships between team members. To translate the graph into a feature representation, six graph centrality kernel functions were applied to represent each node of the team graph. Table 4 describes the kernel functions: (1) PageRank (2) betweenness centrality (3) closeness centrality (4) degree centrality (5) in degree centrality (6) Katz centrality. To create the final team representation, the vectors of individual nodes were averaged, and doc2vec was applied to create the embedding. With eight lexical categories and six kernel functions, the length of the feature vector is 48. This method corresponds to the Kernel version of Entrainment2Vec [20] and achieves comparable performance when applied to the whole dialogue.

Our implementation is slightly different from that of [20] and [19] in two aspects. First, we are using the NLTK POS tagger to assign lexical categories to the utterances instead of using LIWC-derived categories. Second, we are using six graph kernel algorithms instead of ten. The POS tagging reflects the sentence's syntactic structure; we have carefully selected the POS categories that are consistent with the conventional English part of speech categories used by [20] and [19]. While calculating the entrainment, we do not consider the actual word and its context; therefore, this embedding only captures syntactic features, not semantics.

4.4. *Doc2vec*

Le and Mikolov [30] introduced doc2vec as an unsupervised learning algorithm to generate distributed vector representations of text of arbitrary size; it is inspired by the word2vec model [33]. They proposed two different models for learning numerical representations of text: 1) Distributed Memory Model of Paragraph Vectors (PV-DM) 2) paragraph vector with a distributed bag of words (PV-DBOW).

Distributed Memory Model of Paragraph Vectors (PV-DM) uses both word vectors and paragraph vectors to predict the next word. It attempts to learn paragraph vectors that can predict the word given different contexts sampled from the text. The context size is a tuneable parameter, and a sliding window of arbitrary

Table 4: Entrainment Kernel Functions

Kernel Function	Description
PageRank	Ranks the node based on the quality and number of incoming links
Betweenness centrality	Measures the centrality of the node based on the shortest paths (measures information flow)
Closeness centrality	Reciprocal of the sum of the length of the shortest paths between the node and the rest of the graph (measures efficiency of information spread)
Degree centrality	Number of incoming and outgoing entrainment connections
In-degree centrality	Number of incoming entrainment connections only
Katz centrality	Measures the number of walks between two nodes, reflecting its relative influence on neighbors.

context size generates multiple context samples. Doc2vec works by averaging these word vectors and paragraph vectors to predict the next word. It employs stochastic gradient descent to learn word and paragraph vectors. The resultant paragraph vectors serve as a feature vector of the corresponding paragraph and can be used as an input to machine learning models like SVM and logistic regression.

Paragraph vector with a distributed bag of words (PV-DBOW) ignores the context words and attempts to predict randomly selected words from the paragraph. At each iteration of stochastic gradient descent, it classifies a randomly selected word from the sampled text window using paragraph vectors.

Instead of using doc2vec on the raw team dialogues, doc2vec was applied to the output of the dialogue act classifier, sentiment analysis, and syntactic entrainment. This procedure enables us to disentangle the contribution of different elements of team communication at predicting conflict.

5. Datasets

This article includes results from three datasets: 1) multiplayer cooperative board games (Teams corpus) [12]; 2) software engineering teams (GitHub issue comments); and 3) military team communications [34]. We test the generalizability of the embeddings learned on the Teams corpus on the two datasets collected from software engineering and military teams. The Teams corpus is the most complete dataset since it is the only one that contains post-task process conflict ratings.

5.1. *Multiplayer Board Games (Teams Corpus)*

We initially apply our proposed methodology on the Teams corpus dataset collected by Litman et al. [12]. It contains 124 team dialogues from 62 different teams, playing two different collaborative board games. The length of the dialogues varies from

291 to 2124 utterances. In addition to collecting dialogue data, the researchers administered surveys of team level social outcomes. Team social outcome scores include task conflict, relation conflict, and process conflict scores. All these scores are highly correlated, and we are using process conflict z-scores to represent team conflict. Jehn et al. have identified that low process conflict scores indicate good team performance and vice versa [35]. To study the problem of early prediction of team conflict, we divide each dialogue into three equal sections that correspond to the knowledge-building, problem solving, and culmination teamwork phases. Our final classification dataset consists of 12 patterns per dialogue, which are generated from applying the three methods (semantic, sentiment, syntactic) to the whole time period, as well as the initial, middle and final segments.

5.2. *Software Engineering Teams (GitHub Issue Comments)*

The GitHub social coding platform is specialized to support virtual teams of software developers whose primary communication goal is to discuss new features and monitor software bugs. Our assumption is that each software repository is maintained by a team and that the events associated with the repository form a partial history of the team activities and social interactions. Within GitHub's issue handling infrastructure, users can report a bug or provide a feature request by opening an issue.

We created a dataset from software engineering teams resolving issues on GitHub which we are in the process of making publicly available at: `https://drive.google.com/file/d/17W3zeyN3EUJAMYTJVbDcPXmg6DQcqXT6/view`. Table 5 shows the statistics of our corpus. The length of the dialogues in our GitHub corpus varies from 2 to 207 utterances. Utterances from the GitHub dialogues, unlike the Teams corpus, are combination of English language words, special symbols, and code written in different programming languages. The average length of the dialogues is 19. The number of speakers varies from 2 to 10. While collecting the dialogues, to preserve the complex nature of the GitHub dialogue we didn't place any limitation on the total number of speakers and the length of the dialogue. Code blocks were removed if they appeared separately in the dialogue but not if they appeared within the utterance.

Table 6 shows the example from GitHub corpus. Since we lack post-task process conflict survey scores from the team members, we manually labeled the dialogues as being high conflict or low conflict using the following criteria:

(1) The issue did not resolve successfully.
(2) The question(s) of the team member(s) remained unanswered.
(3) One or more team members did not understand the issue.
(4) Lack of understanding or disagreement between the team members.
(5) At least one team member did not agree with the suggested solution.

This criteria are based on Kalia et al's [34] work on affective processes in teams. An affective process represents the motivational and affective relationships between

Table 5: Statistics of GitHub Issue Comments Dataset

#Dialogues	# Utterances	#Tokens	#DA tags	#Positive Samples	#Negative Samples
50	981	13418	42	29	21

Table 6: GitHub Dataset Sample

Speaker	Utterance	DA
m1	I'm following up on this SO question as no one else has. The comments recommend posting a feature request here.	sd
m1	I have an R package on github. This R package has C++ dependencies which I include in a src.	sd
m1	The correct way I would normally do this (outside of R) is create submodules within the github repo which could link to the correct commits as dependencies.	sd
m1	So the checking for empty or unneeded directories causes the errors because the submodules are interpreted as empty subdirectories. Therefore it cannot find the necessary dependencies and I'll run into a fatal error upon build	sd
m1	Yes one way to solve this is to physically put the dependencies within the R package. That does defeat the purpose of submodules though which are very useful.	aa
m1	It appears using the following argument works:	sd
m1	The problem with this is this isn't default behavior. I'm nervous about getting dozens of github issues from users who ran$devtools :: install_git("reponamepackagename")$ and didn't read the fine print in the README	sd
m1	Is there a better way?	qy
m1	What is the standard method of releasing R packages as a github repo using submodules?	qw
m2	FWIW there is a on-going PR for installing github repo with submodules in#103. When it is done it may answer your use case.	sv
m3	I would recommend using subtrees instead of submodules which will just work for users without any additional tooling.	sd
m3	As of 0927172 remotes now automatically detects submodules and installs them as needed.	sd

the members of the team. They evaluated dyadic communication between team members including 1) responses to questions, 2) responses to directives, 3) responses to requests, 4) responses to commissives and 5) responses to informatives. The team member's response (taking the required action) to the other team member's directives and requests is an example of positive evidence indicating low conflict. The absence of the response counts as negative evidence. Response to the informatives, questions, and commissives is an example of neutral evidence.

5.3. *Military Dataset*

We also used Kalia et al.'s military team communication dataset [34] which contains 22 chats from 20 chat rooms. The chats are communication from simulation activity (SIMEX). The average number of speakers in their corpus is 15, which is larger than the other two datasets. The length of the dialogue varies from 55 to 1027 utterances. Table 7 shows example utterances from the military dataset and their dialogue act classification. This dataset also contains post-event survey reflecting qualitative measures of team performance. Kalia et al. [34] used the meaning of the messages

Table 7: Military Dataset Sample

Speaker	Utterance	DA
m1	it says 34 cdr is talking in bde room.	sd
m1	bandit 6 came in pretty quiet in bde room	sd
m2	roger	b
m3	are we atlking this T72 spotted by B Co 2-44 IVO 12SWG 61768 89877?	qy
m2	not tracking this one.	sd
m3	2-44 taking small arms fire in and around Airfield West, and from OBJ 5.	sd
m3	WPNs CO 2-44 engaging tech vechicle IVO 12SWG 61794 90137 and 7 dismounts	sd
m1	CTRP 100%.	sd
m2	roger c trp .	sd

Table 8: Doc2Vec Comparison

	PV-DBOW		PV-DM	
	Accuracy	F1-Score	Accuracy	F1-Score
Dialogue Act	57.89	58.25	**68.42**	**68.77**
Sentiment	55.26	55.48	**78.94**	**77.53**
Entrainment	55.26	55.04	**60.52**	**60.77**

from broadcast communication to evaluate how the team process measures change with time; we use the post-event survey results to annotate the whole teamwork chat as being high or low conflict.

6. Experimental Setup

Teams were divided into low and high conflict teams based on their process conflict z-scores, and classification accuracy was measured. Doc2vec was used to generate the vector representation of all the patterns. Doc2vec comes in two different flavors: 1) Distributed Memory Model of Paragraph Vectors (PV-DM) and 2) Distributed Bag of Words version of Paragraph Vector (PV-DBOW). Table 8 shows the comparison of PV-DM & PV-DBOW when applied to the complete dialogue. The main difference between PV-DM and PV-DBOW is, unlike PV-DBOW, PV-DM keeps track of the context while encoding. The high performance of PV-DM on DAs and sentiment patterns, compared to PV-DBOW, confirm that the sequences contain meaningful information. Our results show that the performance difference of the PV-DM and PV-DBOW using the dialogue act and sentiment embeddings are statistically significant ($p < .01$). The difference of the PV-DM and PV-DBOW using the entrainment embedding is not statistically significant (p=0.754). PV-DM gives consistent performance across all the three features sets, making it a better candidate for detailed analysis. Through extensive experiments, we identified that PV-DM with epoch size of 5, negative sampling 5, and window size 10 works best for our setting. By default, we only report results for PV-DM.

We evaluated the performance of both logistic regression and the support vector machine (SVM) classifier on the full dialogue (shown in Table 9). SVM clearly

Table 9: Comparison of Supervised Classifiers

	Logistic Regression		SVM	
	Accuracy	F1-Score	Accuracy	F1-Score
Dialogue Act	63.15	63.15	**68.42**	**68.77**
Sentiment	71.05	70.86	**78.94**	**77.53**
Entrainment	**63.15**	**63.15**	60.52	60.77

Table 10: Accuracy by Team Phase

Phase	Dialogue Act		Sentiment		Entrainmemt	
	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score
Whole	68.42	68.77	**78.94**	**77.53**	60.52	60.77
Initial	**71.05**	**71.35**	65.78	62.84	42.10	42.42
Middle	**73.68**	**73.31**	65.78	59.18	47.36	46.78
End	68.42	68.68	**71.05**	**71.19**	60.52	60.32

performed better than logistic regression using the dialogue act and sentiment embeddings. Logistic regression seemed to perform better on entrainment compared to the SVM. We report the detailed comparison of the two classifiers by incrementally increasing the length of the dialogues in Section 7. To remain consistent with the previous work [20], SVM was used for the teamwork phase comparison.

7. Results on Teams Corpus

Table 10 presents the classification accuracy of the three embeddings on the whole dialogue. SVM exhibits the best classification accuracy of 78.94% on sentiment based vectors, followed by dialogue act based vectors. Figure 2 visually illustrates the effects of different embeddings. By plotting the vectors in 2d using t-Distributed Stochastic Neighbor Embedding (TSNE), we can observe the formation of two clusters, representing teams with high social outcomes and low social outcomes in the dialogue act and sentiment vectors, whereas the entrainment ones are intermixed. Table 10 shows the accuracy of the conflict classifier across the duration of the games. The sentiment classifier achieved the best accuracy when the whole dialogue was used and exhibited consistent performance across all team phases. The dialogue act embedding was the best at the initial phase, making it a good choice for the "thin-slice" problem of rapidly diagnosing teamwork health from a small sample of utterances. Syntactic entrainment lagged behind the sentiment and semantic analysis, but performance improved during the final phase. Note that each phase was analyzed separately, rather than cumulatively.

For statistical testing, we generated 30 results for each phase using each embedding. Since some of the result distributions (Figure 3) failed the D'Agostino-Pearson normality test, the Kolmogorov-Smirnov test was used for significance testing. The performance differences between each pair of embeddings were statistically significant ($p < 0.01$). However the differences between the initial and end phase results for the sentiment and entrainment embeddings were not significant (Table 11). Se-

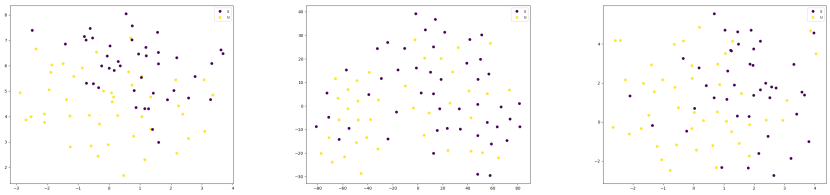

Fig. 2: t-SNE representation of vectors in 2D, where 'S' represents the teams with low process conflict scores and 'U' represents the teams with high process conflict scores. Both sentiment (left) and dialogue act embedding (right) show a better class separation than entrainment (center). Note that the axes have no explicit meaning.

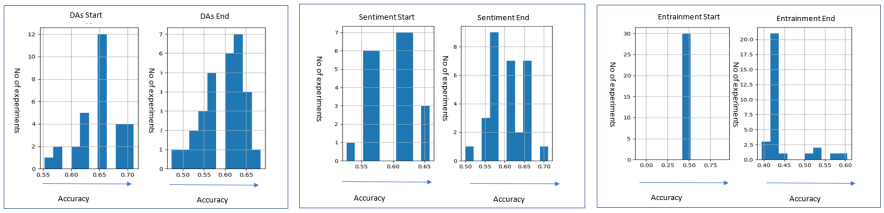

Fig. 3: Distribution of embedding results for initial and final teamwork phases for dialogue acts (left), sentiment (middle) and entrainment (right)

Table 11: Comparison of approaches during the initial (knowledge discovery) and culmination (final) phases

	Knowledge Discovery		Culmination		
	min	max	min	max	p-value
Dialogue Act	0.552632	0.710526	0.473684	0.684211	**2.48e-05**
Sentiment	0.526316	0.657895	0.500000	0.710526	0.455695
Entrainment	0.4210	0.4210	0.394737	0.605263	0.594071

mantic and sentiment based vectors outperformed the syntactic entrainment vectors at the classification task across all phases.

Preliminary results (Table 9) showed that entrainment vectors perform slightly better when used with logistic regression than with SVM. To further analyze the finding and test our third hypotheses (**H3**), we check the embeddings' performance as the dialogue progresses. For this purpose, we divide the dialogues into 20 phases. Starting from the first phase of the dialogue, we incrementally increase the dialogue's length by adding the next phase into it. This is different from testing on knowledge building phase, problem-solving phase, and culmination phase, where while training and testing on any specific phase, we did not include utterances from previous phases. Figure 4 shows the trend of classification performance of the dif-

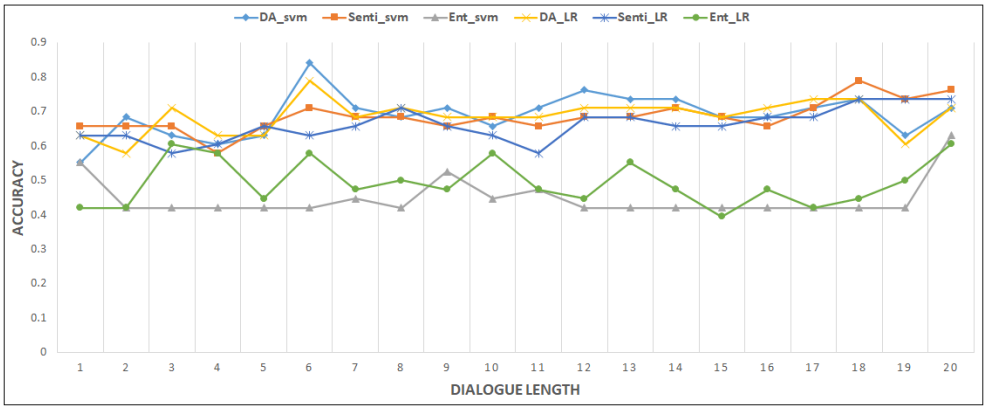

Fig. 4: Conflict prediction accuracy of different embeddings on the Teams corpus as the dialogue progresses. The classifiers (SVM and logistic regression) using the entrainment embedding (green and gray lines) perform consistently worse across the whole dialogue.

ferent embeddings with both logistic regression and SVM classifiers as the dialogue progresses. Results showed that both sentiment and dialogue act embeddings dominant across the whole timeframe. The results also reject **H3** by showing that the entrainment performance does not improve as the dialogue progresses.

8. Results on Dataset Generalization

One of our research goals is to create team communication embeddings that generalize well across datasets, since there are few team communication datasets and some of them are extremely small. To evaluate generalizability, we apply our pre-trained models (dialogue acts, sentiment, entrainment) on the GitHub issue comments and military dialogues.

Table 12 shows the performance of the different embeddings on the GitHub issue comments. The dialogue act embedding outperforms the other ones under both classifiers and achieves a comparable performance to the original dataset. The dialogue act embedding also outperforms sentiment on the small military team communication dataset (Table 13). Unfortunately, the pre-trained entrainment embedding completely failed on this problem. One issue with the military dataset is that it features significantly larger teams (15 members on average) than the Teams corpus (3-4 members). We believe that graph based entrainment measures do not generalize well across larger graphs since the kernel measures are very dependent on graph size. Also the length of dialogues in GitHub issue comments is short compared to the Teams corpus; many team members only have one utterance in a dialogue. The small number of utterances from a team member doesn't facilitate effective

Table 12: Performance on GitHub Issue Comments Dataset

	Logistic Regression	SVM
Dialogue Act	**66.00**	**68.00**
Sentiment	58.00	60.00
Entrainment	42.00	42.00

Table 13: Performance on Military Teams Dataset

	Logistic Regression	SVM
Dialogue Act	**100.00**	**100.00**
Sentiment	90.00	60.00
Entrainment	-	-

computation of entrainment.

8.1. *Improving Conflict Detection Performance*

Our long-term goal is to create a proactive assistant agent that can rapidly detect team conflicts using the dialogue act embedding. To do this, we want to maximize the F1-score of the high conflict class (unsuccessful teams). Fine-tuning the model on the target dataset is one way to improve the pre-trained model's performance on the target dataset. Due to the small size of the Teams corpus, we do not use an extensive deep learning model; to analyze the performance of the classifier when samples from the target dataset are used for training along with the actual training corpus, we add five high conflict dialogues from the GitHub issue comments dataset to the training dataset. This is not possible to do with the military dataset which lacks good examples of conflict. We have intentionally selected a minimal number of samples from the target dataset to avoid cheating the generalizability check. Table 14 shows the comparison of F1-scores of the individual classes when the dialogue act embedding is trained with and without supplemental high conflict examples. The GitHub dataset contains 50 dialogues, of which we are using 5 dialogues for training and 45 for testing. Incorporating GitHub high conflict samples in training the dialogue act embedding also improved the accuracy of the low conflict class. We were able to use a similar approach to boost the performance of the sentiment embedding, but the final performance remained lower than the dialogue act embedding.

9. Conclusion and Future Work

This study presents an evaluation of different embeddings for predicting team conflict from multiparty dialogue. Embeddings were extracted from three types of features: 1) dialogue acts 2) sentiment polarity 3) syntactic entrainment. Results confirm the effectiveness of both sentiment (**H2**) and dialogue acts (**H1**). However, experiments failed to confirm that classification based on syntactic entrainment

Table 14: Performance on GitHub Issues Dataset With vs. Without High Conflict Training Examples

SVM Classifier			
	Accuracy (overall)	Low Conflict (F1-Score)	High Conflict (F1-Score)
Without	68.88	80.00	30.00
With	**73.33**	**81.00**	**54.00**
Logistic Regression Classifier			
	Accuracy (overall)	Low Conflict (F1-Score)	High Conflict (F1-Score)
Without	71.11	81.00	38.00
With	**75.55**	**84.00**	**52.00**

signficantly improves over time (**H3**). Although there are many other ways to measure linguistic synchronizaton, it seems less promising for integration into an agent assistance system. The dialogue act embedding is strong during the initial phase making it a good candidate for diagnosing the health of team formation activity. A continuous team monitoring agent assistant system might do better with sentiment analysis, assuming training data availability.

The highly specialized nature of the team communication produced by software engineering and military teams make them excellent candidates to evaluate the learned embeddings. We test models trained on the Teams corpus on these other datasets. The dialogue act embedding generalized better than sentiment and entrainment on real-world datasets from software engineers and military teams (**H4**). Results show that fine-tuning on the target dataset improves performance. Sentiment embeddings show some potential but seem more promising when trained and tested on the same corpus. Due to its usage of graph kernels, the entrainment feature vector is highly dependent on consistent team sizes and did not generalize well on either corpus.

In future work we plan to explore embeddings based on macrocognitive teamwork states, such as those in the Macrocognition in Teams Model (MITM) [36]. Drawing from research on externalized cognition, team cognition, group communication and problem solving, and collaborative learning and adaptation, MITM provides a coherent theoretically based conceptualization for understanding complex team processes and how these emerge and change over time. It captures the parallel and iterative processes engaged by teams as they synthesize these components in service of team cognitive processes such as problem solving, decision making and planning. We also plan to study loop closure in teams which has been identified as a key discriminator of team performance by several studies [37, 38]. A closed loop occurs when a message has been sent, received, and acknowledged. We believe this pattern can be detected using our dialogue act classifier and used to provide a good real-time indicator of team performance.

10. Acknowledgement

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. W911NF-20-1-0008. Any opinions,

findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or the University of Central Florida.

References

[1] M. F. Jung, Coupling interactions and performance: Predicting team performance from thin slices of conflict, *ACM Transactions on Computer-Human Interaction (TOCHI)* **23**(3) 1–32 (2016).
[2] N. Ambady and R. Rosenthal, Thin slices of expressive behavior as predictors of interpersonal consequences: a meta-analysis, *Psycholology Bulletin 111* **2** 256—274 (1992).
[3] N. Ambady and R. Rosenthal, Half a minute: predicting teacher evaluations from thin slices of non-verbal behavior and physical attractiveness, *J. Pers. Soc. Psychol.* **64**(3) 431—441 (1993).
[4] A. Tekleab, N. Quigley and P. Tesluk, A longitudinal study of team conflict, conflict management, cohesion, and team effectiveness, *Group and Organization Management* **34**(2) 170–205 (2009).
[5] K. Jehn, A multimethod examination of the benefits and determinants of intragroup conflict, *Administrative Science Quarterly* **40** 256–282.
[6] C.-W. Goo and Y.-N. Chen, Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts, in *IEEE Spoken Language Technology Workshop (SLT)* 2018, pp. 735–742.
[7] D. Reitter and J. D. Moore, Predicting success in dialogue, *Proceedings of the ACL* (2007).
[8] C. Danescu-Niculescu-Mizil, M. Gamon and S. Dumais, Mark my words!: linguistic style accommodation in social media, in *Proceedings of the International Conference on World Wide Web*, 2011, pp. 745—754.
[9] J. S. Pardo, On phonetic convergence during conversational interaction, *The Journal of the Acoustical Society of America* **119**(4) 2382—2393 (2006).
[10] M. Mizukami, K. Yoshino, G. Neubig, D. Traum and S. Nakamura, Analyzing the effect of entrainment on dialogue acts, in *Proceedings of the Annual Meeting of the Special Interest Group on Discourse and Dialogue* (Association for Computational Linguistics, Los Angeles, September 2016), pp. 310–318.
[11] Y. Yang, G. N. Kuria and D.-X. Gu, Mediating role of trust between leader communication style and subordinate's work outcomes in project teams, *Engineering Management Journal* **32**(3) 152–165 (2020).
[12] D. Litman, S. Paletz, Z. Rahimi, S. Allegretti and C. Rice, The Teams corpus and entrainment in multi-party spoken dialogues, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing* 2016, pp. 1421–1431.
[13] S. Bird, B. Boguraev, M. Kay, D. McDonald, D. Hindle and Y. Wilks, *Survey of the state of the art in human language technology* (Cambridge University Press, 1997).
[14] S. Parsons, S. Poltrock, H. Bowyer and Y. Tang, Analysis of a recorded team coordination dialogue, in *Proceedings of the Second Annual Conference of the ITA* 2008.
[15] D. N. Walton and E. C. W. Krabbe, *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning* (State University of New York Press, 1995).
[16] J. E. McGrath, Time, interaction, and performance, *Small Group Research* (1991).
[17] G. Sukthankar, K. Sycara, J. A. Giampapa, C. Burnett and A. Preece, An analysis of salient communications for agent support of human teams, in *Multi-agent Systems: Semantics and Dynamics of Organizational Models*, ed. V. Dignum (IGI

Global, 2009), pp. 284–312.
[18] A. Shibani, E. Koh, V. Lai and K. J. Shim, Assessing the language of chat for teamwork dialogue, *Journal of Educational Technology & Society* **20**(2) 224–237 (2017).
[19] C. Danescu-Niculescu-Mizil, L. Lee, B. Pang and J. Kleinberg, Echoes of power: Language effects and power differences in social interaction, in *Proceedings of the International Conference on World Wide Web* 2012, pp. 699–708.
[20] Z. Rahimi and D. Litman, Entrainment2vec: Embedding entrainment for multiparty dialogues, in *Proceedings of the AAAI Conference on Artificial Intelligence* **34**(05)2020, pp. 8681–8688.
[21] K. Sawhney, M. C. Prasetio and S. Paul, Community detection using graph structure and semantic understanding of text, *SNAP Stanford University* (2017).
[22] K. Xu, J. Li and S. S. Liao, Sentiment community detection in social networks, in *Proceedings of the iConference*, 2011, pp. 804–805.
[23] S. Liu, L. Wang, S. Lin, Z. Yang and X. Wang, Analysis and prediction of team performance based on interaction networks, in *Chinese Control Conference (CCC)* (IEEE, 2017), pp. 11250–11255.
[24] F.-S. Yang and C.-H. Chou, Prediction of team performance and members' interaction: A study using neural network, in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (Springer, 2014), pp. 290–300.
[25] M. Omar, S.-L. Syed-Abdullah and N. M. Hussin, Developing a team performance prediction model: A rough sets approach, in *International Conference on Informatics Engineering and Information Science* (Springer, 2011), pp. 691–705.
[26] S. Marlow, C. Lacerenza, J. Paoletti, C. S. Burke and E. Salas, Does team communication represent a one-size-fits-all approach?: A meta-analysis of team communication and performance, *Organization Behavior and Human Decision Processes* **144** 145–170 (2018).
[27] P. Sen and A. Saffari, What do models learn from question answering datasets?, *arXiv preprint arXiv:2004.03490* (2020).
[28] M. Segù, A. Tonioni and F. Tombari, Batch normalization embeddings for deep domain generalization, *arXiv preprint arXiv:2011.12672* (2020).
[29] S. Wang, L. Yu, K. Li, X. Yang, C.-W. Fu and P.-A. Heng, Dofe: Domain-oriented feature embedding for generalizable fundus image segmentation on unseen datasets, *IEEE Transactions on Medical Imaging* **39**(12) 4237–4248 (2020).
[30] Q. Le and T. Mikolov, Distributed representations of sentences and documents, in *International Conference on Machine Learning* (PMLR, 2014), pp. 1188–1196.
[31] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar *et al.*, Universal sentence encoder, *arXiv preprint arXiv:1803.11175* (2018).
[32] Textblob https://textblob.readthedocs.io/en/dev/.
[33] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, in *Advances in Neural Information Processing Systems* 2013, pp. 3111–3119.
[34] A. K. Kalia, N. Buchler, A. DeCostanza and M. P. Singh, Computing team process measures from the structure and content of broadcast collaborative communications, *IEEE Transactions on Computational Social Systems* **4**(2) 26–39 (2017).
[35] K. A. Jehn and E. A. Mannix, The dynamic nature of conflict: A longitudinal study of intragroup conflict and group performance, *Academy of Management Journal* **44**(2) 238–251 (2001).
[36] S. M. Fiore, S.-J. K. A., E. Salas, N. Warner and L. M., Toward an understanding

of macrocognition in teams: Developing and defining complex collaborative processes and products, *Theoretical Issues in Ergonomic Science* **11**(4) 250–271 (2010).

[37] C. Bowers and F. Jentsch, Using communications analysis to understand team development: An example, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* **46**(2) 295–297 (2002).

[38] M. Diaz and K. Dawson, Impact of simulation-based closed-loop communication training on medical errors in a pediatric emergency department, *American Journal of Medical Quality* **35**(6) 474—-478 (2020).