Entrainment2Vec: Embedding Entrainment for Multi-Party Dialogues

Zahra Rahimi, Diane Litman
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA 15260
{zar10, dlitman}@pitt.edu

Abstract

Entrainment is the propensity of speakers to begin behaving like one another in conversation. While most entrainment studies have focused on dyadic interactions, researchers have also started to investigate multi-party conversations. In these studies, multi-party entrainment has typically been estimated by averaging the pairs' entrainment values or by averaging individuals' entrainment to the group. While such multi-party measures utilize the strength of dyadic entrainment, they have not yet exploited different aspects of the dynamics of entrainment relations in multi-party groups. In this paper, utilizing an existing pairwise asymmetric entrainment measure, we propose a novel graph-based vector representation of multi-party entrainment that incorporates both strength and dynamics of pairwise entrainment relations. The proposed kernel approach and weakly-supervised representation learning method show promising results at the downstream task of predicting team outcomes. Also, examining the embedding, we found interesting information about the dynamics of the entrainment relations. For example, teams with more influential members have more process conflict.

1 Introduction

Entrainment is the propensity of speakers to begin behaving like one another in conversation. Evidence of entrainment has been found for multiple aspects of speech, including lexical choice (Brennan and Clark 1996; Metzing and Brennan 2003; Niederhoffer and Pennebaker 2002; Danescu-Niculescu-Mizil, Gamon, and Dumais 2011; Gonzales, Hancock, and Pennebaker 2010; Pennebaker, Francis, and Booth 2001; Beňuš, Levitan, and Hirschberg 2012; Rahimi et al. 2017; Friedberg, Litman, and Paletz 2012) in both human-human and human-computer dialogues. In addition, the strength of entrainment has been shown to be associated with numerous social and conversational qualities, such as the cohesiveness of speech (Lubold and Pon-Barry 2014; Natale 1975; Rahimi and Litman 2018; Beňuš et al. 2014; Danescu-Niculescu-Mizil et al. 2012).

While entrainment studies have largely focused on two-party conversations, recent studies have estimated entrainment in multi-party conversations by averaging pairs' entrainment values, or by averaging individuals' entrainment to the group (Gonzales, Hancock, and Pennebaker 2010; Nenkova, Gravano, and Hirschberg 2008; Friedberg, Litman, and Paletz 2012; Danescu-Niculescu-Mizil et al. 2012; Doyle and Frank 2016; Rahimi and Litman 2018). However, the entrainment structure of multi-party groups consists of many pairwise relations with different strengths and directions. To the best of our knowledge, almost no existing multi-party entrainment measures utilize the structure of pairwise entrainment relations in groups. Measuring convergence, Rahimi and Litman (2018) introduced a weighting based on entrainment behaviors (converging, diverging, or maintaining) in groups to decrease the influence of outliers. Although this is an attempt to utilize the structure of group entrainment relations, it encodes only one aspect of these relations.

In this paper, utilizing the adaptive pairwise entrainment measure (Danescu-Niculescu-Mizil et al. 2012), which measures how much a speaker adapts her language to another one in a local turn-by-turn basis, we propose a graph-based vector representation of multi-party entrainment to encode the strength and structure of pairwise interactions in multi-party groups. Weighted directed entrainment graphs represent the structure of pairwise entrainment relations and their strength. Learning embedding for the entrainment graphs, we represent multi-party entrainment in vector-space where groups with similar graphs have close vectors.

Learning dense vector representations for nodes, edges, or sub-graphs from a large-scale sparse graph has been studied by researchers and applied to social or knowledge graphs (Luo et al. 2015; Tang et al. 2015; Grover and Leskovec 2016; Zhou et al. 2017; Hamilton, Ying, and Leskovec 2017). The intuition is similar to *word2vec* (Mikolov et al. 2013). Similar nodes in a graph should have vector representations that are close to each other. Similarity can be defined as having similar neighbors or similar structural roles.

Inspired by these methods and by *paragraph2vec* (Le and Mikolov 2014), we learn vectors for small graphs where similarity is defined as having a similar graph structure. To encode the structure of small graphs/groups (only about 4 nodes/speakers), we propose to initialize the node and graph embeddings by applying a set of graph algorithms where each encodes a distinctive property of the graph. As the supervision, we propose to employ the domain-specific task of predicting real versus randomly permuted conversations, which has been utilized in the entrainment domain to verify the validity of measures (De Looze et al. 2014; Lee et al. 2011; Jain et al. 2012; Rahimi and Litman 2018). Experimental evaluations demonstrate that the group entrainment embedding improves performance for the downstream task of predicting group outcomes compared to the state-of-the-art methods.

2 Group Entrainment Graphs

Influence networks or graphs have been introduced and investigated in other domains such as the social analysis literature (Friedkin and Johnsen 2011; Tang et al. 2009; Romero et al. 2011). We propose to apply a similar idea to build multi-party entrainment graphs. First, we estimate pairwise entrainment values using an existing probabilistic directional (i.e., asymmetric) method (Danescu-Niculescu-Mizil et al. 2012), where the entrainment of speaker b to speaker a on a lexical category or linguistic feature c, $Ent_c(a, b)$, is defined as in Equation 1 [1]:

$$Ent_c(a,b) = p(e_b|e_a) - p(e_b) \tag{1}$$

Let U_{ab} be the set of all (u_b, u_a) utterance pairs where speaker a's utterance (u_a) immediately precedes speaker b's utterance (u_b). e_b/e_a is the indicator that the desired event (e.g., presence of lexical category c) occurs in the corresponding utterances u_b/u_a from set U_{ab} respectively. The directionality of this entrainment measure means that the entrainment of speaker a towards b is not the same as the entrainment of speaker b towards a.

Next, we define the multi-party entrainment graph $G = (V, E, W)$. Each node $v \in V$ represents a speaker from the group. The directed edges in E represent the presence and the weights in W represent the strength of entrainment between the source and destination node, which are measured using Equation 1. We define an edge from node a to b if and only if $Ent_c(a, b)$ is positive. Equation 1 measures adaptation in language and negative values imply no adaptation. In other words, a negative value implies that speaker b uses the linguistic feature c in response to speaker a more than in response to speaker a's speech with linguistic feature c. So, to simplify the problem, we decide to consider the negative values as no entrainment. There are other pairwise entrainment measures where the sign is more meaningful and cannot be simply ignored, such as convergence (Levitan and Hirschberg 2011). In future work, we will investigate considering both positive and negative values. The direction of the edge implies that the source node influences the destination node, while the edge weight represents the amount of this influence (entrainment). Consider a group with three speakers: A, B, C. Suppose that the entrainment of pairs on a lexical feature are as follows:

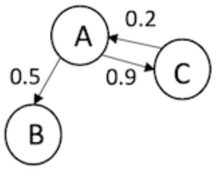

Figure 1: An example multi-party entrainment graph.

$$\begin{aligned} &Ent(A,B) = 0.5, Ent(B,A) = -0.1,\\ &Ent(A,C) = 0.9, Ent(C,A) = 0.2,\\ &Ent(C,B) = -0.5, Ent(B,C) = 0. \end{aligned}$$

The entrainment graph that would then be constructed for this group is shown in Figure 1.

Entrainment graphs contain interesting information about the dynamics of the entrainment relations. For example, we could learn the structural roles of the speakers, such as who are the influencers, connectors, or passive speakers. Also, we could learn about indirect entrainment relations. If A influences B and B influences C, there is an indirect influence from A to C. This information could potentially help us have a better understanding of multi-party entrainment. In the graph from Figure 1, we observe that A is an influential speaker and C could potentially influence B although there is no edge between them. This information will be lost if we simply average pairs' entrainment values.

In the next section, we propose three different approaches to learn entrainment vector representations from such entrainment graphs.

3 From Graphs to Vector Representation

3.1 Directly Estimating the Vectors

Given the entrainment graphs, the goal is to represent group entrainment in a vector-space where groups with similar entrainment dynamics have close vectors. As the most obvious and straightforward approach, we apply a set of graph algorithms to capture distinctive and informative properties of the graphs. Applying d functions, each node in the graph (i.e., conversational participant) is represented with a d-dimensional vector. Then, the vector representation of the entrainment graph (i.e., the group of participants) is a simple average of its nodes' vectors.

Following are all ten kernel functions that we utilized. We tried to be inclusive but we did not perform an experiment to obtain the best list by pruning it or by adding other algorithms. For convenience, we call these functions kernels in the rest of this paper. All these functions are well-known graph algorithms, so we explain them minimally here. Given a weighted directed graph $G = (V, E, W)$, our 10 kernels are as follows:

K1= Closeness Centrality (Wasserman and Faust 1994) of a node u is "the ratio of the fraction of reachable nodes, to the reciprocal average distance from the reachable node":

$$C(u) = \frac{n-1}{N-1}\frac{n-1}{\sum_{v=1}^{n-1} d(v,u)} \tag{2}$$

[1] Defining a new pairwise entrainment measure is not the contribution of this paper. The contribution is to define a new multi-party measure, using an existing state-of-the art (Danescu-Niculescu-Mizil et al. 2012) pairwise measure.

The distance function $d(v, u)$ is defined as the length of the shortest-path from node v to u. And, n is the number of nodes that can reach u. N is the total number of nodes. In our work, we are interested in finding the highly influential nodes which can reach to many nodes. This is opposite of the Closeness Centrality definition. For this purpose, for each entrainment graph G, we build and use the reversed graph G^R in which the edges are the reverse of G's edges. Thus, the reachable nodes of u in G^R are the ones that can be reached starting from u in the original graph G. So, highly influential nodes in G have high closeness centrality score.

K2 = Betweenness Centrality (Brandes 2001) of node u is the fraction of all-pairs shortest paths that pass through u to all the all-pairs shortest paths. $\sigma(s,t)$ is the number of shortest (s,t)-paths, and $\sigma(s,t|u)$ is the number of those paths passing through node u other than s, t. If $s = t$, $\sigma(s,t) = 1$, and if $u \in \{s,t\}, \sigma(s,t|u) = 0$.

$$C_B(u) = \sum_{s,t \in V} \frac{\sigma(s,t|u)}{\sigma(s,t)} \tag{3}$$

A node has a high betweenness centrality if its role in the graph is a connector.

K3 =PageRank (Page et al. 1999) is one of the most well-known algorithms which was originally designed to rank web pages. It outputs a probability distribution for nodes based on the score of their neighbours. "PageRank works by counting the number and quality of links to a node to determine a rough estimate of how important the node is. The underlying assumption is that more important nodes are likely to receive more links from other nodes." We use the weighted reversed graph G^R and the weighted algorithm (Xing and Ghorbani 2004) to measure PageRank probability distribution of the nodes.

K4 , K5 = HITS (Kleinberg 1999) computes two numbers for a node. Authority score is based on the incoming links ("a good authority is a node that is linked by many different hubs"). Hub score is based on outgoing links ("a good hub is a node that points to many other nodes").

K6 = Maximum Flow (Ford and Fulkerson 2009) of node u is the sum of all single-commodity max flow values (i.e., net outflow) from the source node u to all other nodes in the capacity graph $G^c = (V^c, E, C)$ where each edge has only a single attribute, capacity C, which is equal to the weight attribute in the original graph G.

$$MaxFlow(u) = \sum_{v \in V^c} max\_flow(u, v, C) \tag{4}$$

The Maximum Flow is the sum of all direct and indirect entrainment influences that a node has on all other nodes in the graph.

K7 = Katz Centrality (Katz 1953) computes centrality for a node u based on centrality of its neighbors:

$$KATZ_u = \alpha \sum_j A_{ij} KATZ_j + \beta \tag{5}$$

where A is the adjacency matrix of graph G with eigenvalues λ. The parameter β controls the initial centrality and $\alpha < \frac{1}{\lambda_{max}}$. "Katz centrality computes the relative influence of a node within a network by measuring the number of the immediate neighbors and also all other nodes in the network that connect to the node under consideration through these immediate neighbors."

K8 = Weighted In_degree of a node is the sum of the weights of all incoming edges to it. This indicates how much direct influence the node gets from other nodes in the graph.

K9 = In_degree Centrality of a node is the fraction of nodes to which its incoming edges are connected. This measure indicates how many influencers a node has.

K10 = Degree Centrality of a node is the fraction of nodes that it is connected to (sum of in_degree and out_degree centrality). This measure indicates the ratio of all the nodes that entrain directly to/from the corresponding node.

3.2 Learning Embedding: The Self-Supervised Approach

Given the entrainment graphs, the goal is to learn a d-dimensional embedding of nodes and graphs where nodes with similar structural roles and graphs with similar structure have close vectors. We employ the *node2vec* (Grover and Leskovec 2016) and *paragraph2vec* (Le and Mikolov 2014) methods and define our problem as follows.

Our embedding learning is a maximum likelihood optimization. We define G as the set of all graphs and U as the set of all nodes (vocabulary) from all graphs. Then, for a given graph $g = (V, E, W)$, the nodes of the graph, V, are its context, $C(g) = V$.[2] We seek to optimize the objective function in Equation 6 which maximizes the log-probability of observing the context of the graph g, conditioned on vector representation of g, given by $f(g)$:

$$max_f \sum_{g \in G} \log p(C(g)|f(g)) \tag{6}$$

Assuming conditional independence of observing each node in the context given the vector representation of the graph, the conditional probability of Equation 6 is defined by:

$$P(C(g)|f(g)) = \prod_{v \in C(g)} p(v|f(g)) \tag{7}$$

Let W be the matrix of graph embedding and Z be the matrix of node embedding. Every unique graph g is mapped to a unique vector W_g and every unique node is mapped to a unique vector Z_v. The probability in Equation 7 is defined as the softmax probability normalized by all the nodes in U:

$$p(v|f(g)) = \frac{\exp(Z_v^T.W_g)}{\sum_{u \in U} \exp(Z_u^T.W_g)} \tag{8}$$

Given Equations 6, 7, and 8, and approximating the normalization of the softmax, which is a sum over all nodes of all graphs, with negative sampling (Mikolov et al. 2013; Grover and Leskovec 2016). the loss function of the optimization problem is:

[2] The term context is used to better understand our approach with respect to (Le and Mikolov 2014).

$$L = \sum_{g \in G} \sum_{v \in C(g)} (-\log \sigma(Z_v^T.W_g) + \sum_{u \in C'(g)} \log \sigma(Z_u^T.W_g)) \tag{9}$$

The dot product measures the similarity of a node and a graph in the vector-space. For all nodes in the context of a graph, the vector representations of the nodes and the graph should be close in the vector space. $C'(g)$ is a set of random nodes which do not belong to the graph g. σ is the sigmoid function.

Initializing Embedding: As discussed before, we want graphs with similar structure and nodes with similar structural roles to be close in vector space. To achieve this, we need to encode these structures in the vectors. Approaches like random walks (Grover and Leskovec 2016) are not appropriate for such small graphs of about 4 nodes. For this purpose, we utilize the proposed kernel approach from Section 3.1 to initialize the node and graph embedding matrices. So, the vector of each node or graph indicates their structural properties and similarity of vectors indicates similarity of their structures.

Domain-Specific Negative Sampling: To build the set $C'(g)$ in Equation 9, one approach similar to *word2vec* is to randomly sample nodes that are not in the context of the graph (i.e., that are from other graphs). But, this might not be a good approach for our problem as several graphs might have the same structure. So, randomly choosing nodes from other graphs does not serve our purpose well. We want our negative samples to have different vector representations from the context of the graph.

Entrainment is a phenomena that occurs in the course of conversation. So, randomly generated conversations should not show strong entrainment relations. Distinguishing between real and randomly permuted fake conversations is a validation task in the entrainment literature (De Looze et al. 2014; Lee et al. 2011; Jain et al. 2012; Rahimi et al. 2017). We thus propose to use the permuted version of each conversation to build the corresponding fake graphs and use the nodes of these fake graphs to build $C'(g)$. So, the size of $C'(g)$ is equal to $C(g)$ and the nodes in $C'(g)$ are the permuted version of the nodes in $C(g)$. The fake conversations were generated by randomly permuting the speech and silence intervals of each speaker in the group. Using this method, we make sure that the negative and positive samples have different structures. At the same time, we make sure they are not too distinct since we do not change the content of the conversations but only randomly permute the content. For example, the distribution of the lexical categories are the same in both negative and positive samples of a graph.

Network Structure and Algorithm: As in *word2vec* (Mikolov et al. 2013), our network is a shallow two-layer neural net. It has two embedding look-up tables, one for the nodes and one for the graphs. The second (output) layer includes a sigmoid activation function applied on the dot product of the two vectors from the first layer. The input is a pair of node index and graph index. Given the input indexes and the embedding matrices, we look up the two vectors for the given node and graph. Then, we simply get the dot product of the two vectors to calculate their similarity. We apply a sigmoid activation function on the result of the dot product to calculate the probability of the output. This probability is used to predict if the node is from the context of this graph (a positive sample) or is from the corresponding fake graph (a negative sample). So, we have a simple binary prediction and we use binary cross entropy loss function and a stochastic gradient decent optimization algorithm. After completing the training, we get the learned graph embedding matrix as our representation of multi-party entrainment.

3.3 Learning Embedding: The Weakly-Supervised Approach

The self-supervised approach employs the randomly permuted conversations to generate the negative samples for an optimization task that tries to maximize the likelihood of observing the context nodes. A different approach is to directly employ the validation task of real-vs-permuted prediction as supervision. Given vector representations of a node and a graph, we predict if the input is from a real conversation or a fake conversation. We train the graph embedding while we optimize this classifier. We call this method weakly supervised since the supervision is not on the main task of interest which is predicting social outcomes. Similar to the self-supervised approach, we construct a randomly permuted conversation for each real conversation and build its corresponding entrainment graph. Then, unlike the self-supervised approach, we predict if a given input graph is real or permuted. The objective is to minimize a binary cross entropy loss function or maximize the likelihood of observing the data using a gradient descent optimization.

Like before, we employ two embedding matrices for the nodes and the graphs although the size of the graph embedding is doubled (by including the fake graphs). We initialize the two matrices with the kernel approach to encode the structure of the graphs. Assuming that the kernel initialization is good at encoding the underlying structure, to reduce the number of trainable parameters, we fix the node embedding with the initialization, and train the graph embedding. We utilize the same shallow network as in the self-supervised method. After training the graph embedding, the subset of the graph embedding which is from the real graphs represents our entrainment vectors.

4 Experiments and Results

4.1 Data

To evaluate the utility of our entrainment representations, we use them to measure entrainment in the freely available Teams Corpus (Litman et al. 2016). The corpus includes audio files and transcripts for 62 teams of 3 or 4 individuals playing a cooperative board game in two sessions. The total number of transcripts across sessions for all teams is 119[3]. The teams are disjoint in participants.

Individually taken self-reported pre- and post-game surveys are available for both sessions, including: (1) favorable

[3] Not 124 as a few transcripts are not yet available.

social outcome measures (perceptions of cohesion, satisfaction, potency/efficacy and perceptions of shared cognition), and (2) conflict measures (task, process, and relationship conflicts). Following prior works using the Teams corpus (Rahimi and Litman 2018; Yu and Litman 2019), we created a team-level **Favorable** measure by z-scoring and averaging all the highly correlated favorable measures and averaging them for each team. We followed the prior work and z-scored the process conflict measure and averaged it in the groups to construct a team-level **Conflict** measure. Given the small size of data, binary **Favorable** and **Conflict** measures, split at the median, will be used to evaluate the quality of the different entrainment vector representations from Section 3.

4.2 Experimental Setups

For consistency with prior work (Danescu-Niculescu-Mizil et al. 2012; Doyle and Frank 2016), we measure lexical entrainment on eight LIWC-derived categories of function words (Pennebaker, Francis, and Booth 2001) that have little semantic meaning and are more relevant to style than content. These eight categories are: **articles, auxiliary verbs, conjunctions, high-frequency adverbs, impersonal pronouns, personal pronouns, prepositions, and quantifiers** (451 lexemes total). Transcripts are pre-processed before extracting lexical terms by removing punctuation marks, converting all words to lower case, removing noises such as laughter, and removing any part of the transcript indicated as not fully understood by transcribers.

The size of the input data for training the networks is about 6500 instances[4]. We have minimum 100 and maximum 150 epochs and stop early if the training loss is smaller than 0.1 to avoid overfitting. The batch size is set to 20. *RMSProp* optimization algorithm is used with learning rate equal to 0.001. These hyper-parameters are chosen with regards to the small training data size and should be optimized in the future. It should be emphasized that we do not require any manually labeled data for training the self-supervised or weakly-supervised models.

After learning the entrainment embedding, similar to prior works (De Looze et al. 2014; Lee et al. 2011; Jain et al. 2012; Rahimi et al. 2017; Doyle, Yurovsky, and Frank 2016; Lee et al. 2011) which have evaluated entrainment extrinsically in terms of predicting outcomes, we evaluate its utility at the downstream task of predicting Favorable and Conflict team outcomes. We use support vector machines with RBF kernel and perform leave-one-out cross validation. The size of the data for this experiment is 119 since we predict the outcomes for each team and each session. We compare the utility of the proposed embedding with two local adaptive baselines from the literature: SCP (Danescu-Niculescu-Mizil et al. 2012) which is a probabilistic measure and its pairwise version is utilized in this paper, and HAM (Doyle and Frank 2016) which is a generative hierarchical alignment model argued to outperform SCP. The HAM baseline is fit using the same hyper-parameters as in (Doyle and Frank 2016). But, it is fit with 2000 iterations of the sampler (1000 as warm-ups) and four chains since our data is small. The output is a probability distribution for each group. So, we utilize the mean, upper and lower bounds of the 95% highest posterior density.

4.3 Results and Discussion

First, we utilize the entrainment embedding of all 8 lexical categories as features. So, the total number of features is 80 which for 119 instances of data is a lot and might cause our model to overfit. So, it is important to employ feature selection. We employ a model-based feature selection using LASSO. The feature selection is performed inside the cross validation loop, so the number of selected features might be slightly different at each fold. We also set a threshold on minimum number of features to avoid underfitting. The regularization parameter is tuned with regards to this threshold. The threshold itself is tuned in each fold.

The results are in Table 1. The middle part of the table shows the accuracies of the three proposed approaches using their best configurations at two tasks of predicting Favorable and Conflict outcomes using all features or by employing feature selection. Predicting Conflict, the embedding of weakly-supervised approach when utilizing feature selection outperforms the best SCP result significantly and the best HAM result. Feature selection has a greater effect on the proposed approaches since the number of features in these models are more than baselines. SCP has only 8 features and HAM has 24 features. Predicting Favorable outcome, the embedding of weakly-supervised approach with or without feature selection outperforms the best SCP result significantly and the best HAM result with a trending p-value (< 0.1). Comparing the three proposed approaches, the weakly-supervised approach outperforms the other two models on both tasks. This shows that training the embedding improved the initial kernel vectors although the initial kernel vectors, which does not require any training, performs comparable, if not better, to the computationally expensive HAM baseline.

The last part of Table 1 presents other evaluated configurations for the weakly-supervised and the self-supervised approaches. First, we investigate the importance of the kernel initialization for the self-supervised approach by comparing it to a default **Uniform** initialization. Results in Table 1 show that the two Self_Kernel approaches outperform the Self_Uniform approaches which indicates the importance of a good initialization.

Second, for the self-supervised approach with negative sampling, we evaluate the effect of the proposed negative sampling approach. So, we compare our proposed domain-specific **Permuted** negative samples (self_*_Permuted) with a **Random** negative sampling approach (self_*_Random) where we select random nodes from other graphs rather than from paired permuted graphs. The results in Table 1 show that the proposed negative sampling approach outperforms the random configuration.

For the weakly-supervised approach, the best configuration has Not Trainable (**NT**) node embedding and only trains the graph embedding rather than **T**raining both embedding matrices. This is beneficial since it reduces the number of

[4] 119 sessions * (3 or 4) speakers * 8 LIWC categories * 2 (fake or real)

	Features	Conflict-All	Conflict-FS	Favorable-All	Favorable-FS
Baselines	Majority	53.78	53.78	50.42	50.42
	SCP	63.02	63.86	51.26	50.42
	HAM	68.90	69.74	53.78	51.26
Best Proposed	Kernel	57.14	69.74	$\mathbf{62.18}^{(*,)}$	$\mathbf{62.18}^{(*,)}$
	WeakS_Kernel_Permuted_NT	59.66	$\mathbf{74.79}^{(*,)}$	$\mathbf{63.86}^{(*,+)}$	$\mathbf{63.86}^{(*,+)}$
	Self_Kernel_Permuted	63.02	$\mathbf{73.95}^{(+,)}$	57.14	58.82
Other Configurations	WeakS_Kernel_Permuted_T	58.82	63.86	56.30	57.14
	Self_Kernel_Random	63.02	66.38	56.30	55.46
	Self_Uniform_Permuted	49.58	63.86	39.49	52.94
	Self_Uniform_Random	52.94	57.93	58.82	55.46

Table 1: Accuracy of predicting Conflict and Favorable outcomes. The features are entrainment values/vectors from all 8 LIWC categories. The pair of signs in parenthesis indicates the result of significant test comparing the corresponding accuracies with the best of SCP and HAM in order. "*" indicates significance (p-value < 0.05), "+" indicates trending result (p-value < 0.1).

Task	Features	ipron	article	auxverb	conj	adverb	ppron	preps	quant
Conflict	HAM	**68.06**	**69.74**	63.86	**68.06**	**61.34**	**67.22**	**64.70**	65.54
	Kernel	62.18	63.86	58.82	64.70	**61.34**	**66.38**	60.50	65.54
	WeakS_Kernel_Permuted_NT	57.98	57.14	**65.54**	**67.23**	56.30	60.50	59.66	**72.27**
	Self_Kernel_Permuted	**68.06**	54.62	63.02	57.14	58.82	53.78	55.46	**71.42**
Favorable	HAM	50.42	54.62	45.37	43.69	40.33	47.05	37.81	51.26
	Kernel	**63.02**	63.02	**61.34**	58.82	**55.46**	**52.94**	**63.02**	54.62
	WeakS_Kernel_Permuted_NT	58.82	**66.39**	57.14	57.14	**55.46**	51.26	**63.03**	**57.98**
	Self_Kernel_Permuted	57.14	53.78	48.73	**63.86**	54.62	51.26	47.05	49.58

Table 2: Accuracy of predicting Conflict and Favorable outcomes. The features are entrainment values/vectors from a single lexical category.

trainable parameters of the model. Also, we employ the best initialization method from self_supervised experiments (kernel initialization). Also, the weakly supervised approach does not include negative sampling. But, the task is to predict whether a graph is real or fake.

In a second experiment, we predict the Favorable and Conflict outcomes using the embedding of each lexical category. So, we have 8 prediction tasks for each outcome. At each prediction task, the number of features is equal to the size of the vector (10). So, we do not need to employ feature selections in this experiment. The results are in Table 2. We only experiment with the best methods from the first experiment. For predicting Conflict, we observe that the HAM baseline is more robust across different lexical categories than the embedding approaches. But, the proposed approaches outperform the HAM baseline on all categories when predicting favorable outcome. So, the entrainment embedding is more robust across the two tasks.

Given the promising results of the proposed vector representations, there are two more questions that we further investigate. First, does the proposed vector representation of entrainment learn anything beyond team size? [5] Second, which dimensions (kernels) are more predictive? To answer these questions, we take a closer look at each dimension of the kernel vectors. In the future, we need to expand this experiment and investigate the outperforming weakly-supervised embedding to better understand why learned embedding outperforms the estimated kernels. We pick one of the experiments where the kernel vector representation outperformed baselines on a individual category: kernel model predicting process conflict on the LIWC category of "quantitative". We perform a hierarchical regression analysis. The z-scored team-level process Conflict is the dependent variable. We enter team-size as an independent variable to the first level. In the second level, we add all ten dimensions of the entrainment vector on the LIWC category of *quantitative*.

The results are in Table 3. We remove all the dimensions that did not have a significant or trending coefficient from the final model. The amount of variance explained by Kernel dimensions is significantly above and beyond team size entered in Model 1, $\Delta R^2 = 0.216$, $\Delta F(4, 113) = 8.763$, $p = 0.000$. So, the answer to our first question is yes. The proposed model learns predictive dimensions above and beyond the effect of team size. To answer the second question, we look at the selected dimensions with significant coefficients. Closeness Centrality has a positive significant correlation with process conflict which means teams with higher average closeness centrality have higher conflict. In other words, teams with fewer influential members, have less conflict. Maximum flow has a significant negative correlation which indicates teams with higher average maximum flow, have less conflict. In other words, the more is the direct and

[5] We specially need to answer this question for the Kernel approach, since average of PageRank and HITS scores over the nodes of the graph is the ratio of the number of nodes. This is not an issue for the learned embedding.

	Model 1			Model 2		
	B	SEB	β	B	SEB	β
Team Size	0.613	0.180	0.300*	0.510	0.210	0.250*
K1 = Closeness Centrality				18.922	9.201	4.509*
K2 = Betweenness Centrality				-5.942	3.563	-0.526+
K6 = Maximum Flow				-4.280	1.763	-0.268*
K10 = Degree Centrality				-8.281	4.588	-3.584+
Model R^2		0.09			0.306	
Model F		11.609*			9.948*	

Table 3: Summary of hierarchical regression analysis for variables predicting process conflict on quantitatives using Kernel approach. B , SEB, and β are the unstandardized coefficients, coefficients Std. Error, and standardized coefficients. * $p < .05$. + $p < .1$ $n = 119$.

indirect entrainment in the team, the less is the process conflict. One main advantage of the proposed vector representation compared to the baselines is the ability to present all these pieces of information about the dynamics of entrainment in groups.

5 Conclusion and Future Work

In this paper, we proposed group entrainment embedding, a vector representation for multi-party entrainment to encode the underlying entrainment dynamics in the groups. We proposed three approaches to learn the vector representation from entrainment graphs built by utilizing existing directional pairwise entrainment measures. We concluded that the vector representation learned by our proposed weakly-supervised approach outperforms the baselines and the other proposed approaches. Beside performance, this approach has other advantages. First, encoding the underlying structure of the entrainment graphs in the vectors provides useful information. For example, we found that teams with more influential (in terms of entrainment) members or higher average closeness centrality have more Process Conflict. Second, proposed approaches are computationally less expensive than the best performing baseline: the generative HAM model. Finally, the weakly supervised approach requires training data similar to HAM. But, the proposed kernel approach when performance is comparable to the weakly supervised approach does not require any training data and directly estimates entrainment of groups.

We did not perform any parameter tuning on the parameters of the neural network such as number of epochs, batch size, or learning rate. We also chose a simple two layer network. Other network structures might perform better especially for the weakly-supervised approach. Further investigation is required to optimize the list of the graph algorithms (kernels) to best encode the structure of the graphs. We will also investigate incorporating the Graph Convolutional Networks to encode the graphs. Some other future directions are investigating addition of negative edges to the graphs, understanding the embedding dimensions, utilizing other linguistic features, and incorporating other pair-level entrainment measures. In general, the results are promising and might be even further improved by exploring these paths. Since the Teams corpus is small, the proposed approaches should be validated on a bigger dataset. Also, the team size in Teams corpus is three or four. Larger groups might benefit more from the proposed approaches.

6 Acknowledgements

This work is supported by the National Science Foundation under Grant Nos. 1420784 and 1420377. We thank Omid Kashefi for his valuable comments.

References

Beňuš, Š.; Gravano, A.; Levitan, R.; Levitan, S. I.; Willson, L.; and Hirschberg, J. 2014. Entrainment, dominance and alliance in supreme court hearings. *Knowledge-Based Systems* 71:3–14.

Beňuš, Š.; Levitan, R.; and Hirschberg, J. 2012. Entrainment in spontaneous speech: the case of filled pauses in supreme court hearings. In *Cognitive Infocommunications (CogInfoCom), 2012 IEEE 3rd International Conference on*, 793–797. IEEE.

Brandes, U. 2001. A faster algorithm for betweenness centrality. *Journal of mathematical sociology* 25(2):163–177.

Brennan, S. E., and Clark, H. H. 1996. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 22(6):1482.

Danescu-Niculescu-Mizil, C.; Lee, L.; Pang, B.; and Kleinberg, J. 2012. Echoes of power: Language effects and power differences in social interaction. In *Proceedings of WWW*, 699–708.

Danescu-Niculescu-Mizil, C.; Gamon, M.; and Dumais, S. 2011. Mark my words!: linguistic style accommodation in social media. In *Proceedings of the 20th international conference on World wide web*, 745–754.

De Looze, C.; Scherer, S.; Vaughan, B.; and Cambpell, N. 2014. Investigating automatic measurements of prosodic accommodation and its dynamics in social interaction. *Speech Communication* 58:11–34.

Doyle, G., and Frank, M. C. 2016. Investigating the sources of linguistic alignment in conversation. In *ACL (1)*.

Doyle, G.; Yurovsky, D.; and Frank, M. C. 2016. A robust framework for estimating linguistic alignment in twitter conversations. In *Proceedings of the 25th international conference on world wide web*, 637–648. International World Wide Web Conferences Steering Committee.

Ford, L. R., and Fulkerson, D. R. 2009. Maximal flow through a network. In *Classic papers in combinatorics*. Springer. 243–248.

Friedberg, H.; Litman, D.; and Paletz, S. B. F. 2012. Lexical entrainment and success in student engineering groups. In *Proceedings Fourth IEEE Workshop on Spoken Language Technology (SLT)*.

Friedkin, N. E., and Johnsen, E. C. 2011. *Social influence network theory: A sociological examination of small group dynamics*, volume 33. Cambridge University Press.

Gonzales, A. L.; Hancock, J. T.; and Pennebaker, J. W. 2010. Language style matching as a predictor of social dynamics in small groups. *Communication Research* 37:3–19.

Grover, A., and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 855–864. ACM.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*.

Jain, M.; McDonough, J. W.; Gweon, G.; Raj, B.; and Rosé, C. P. 2012. An unsupervised dynamic bayesian network approach to measuring speech style accommodation. In *EACL*, 787–797.

Katz, L. 1953. A new status index derived from sociometric analysis. *Psychometrika* 18(1):39–43.

Kleinberg, J. M. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)* 46(5):604–632.

Le, Q., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, 1188–1196.

Lee, C.-C.; Katsamanis, A.; Black, M. P.; Baucom, B. R.; Georgiou, P. G.; and Narayanan, S. 2011. An analysis of pca-based vocal entrainment measures in married couples' affective spoken interactions. In *INTERSPEECH*, 3101–3104.

Levitan, R., and Hirschberg, J. 2011. Measuring acoustic-prosodic entrainment with respect to multiple levels and dimensions. In *Interspeech*.

Litman, D.; Paletz, S.; Rahimi, Z.; Allegretti, S.; and Rice, C. 2016. The teams corpus and entrainment in multi-party spoken dialogues. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1421–1431.

Lubold, N., and Pon-Barry, H. 2014. Acoustic-prosodic entrainment and rapport in collaborative learning dialogues. In *Proceedings of the 2014 ACM workshop on Multimodal Learning Analytics Workshop and Grand Challenge*, 5–12. ACM.

Luo, Y.; Wang, Q.; Wang, B.; and Guo, L. 2015. Context-dependent knowledge graph embedding. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1656–1661.

Metzing, C., and Brennan, S. E. 2003. When conceptual pacts are broken: Partner-specific effects on the comprehension of referring expressions. *Journal of Memory and Language* 49(2):201–213.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Natale, M. 1975. Convergence of mean vocal intensity in dyadic communication as a function of social desirability. *Journal of Personality and Social Psychology* 32(5):790.

Nenkova, A.; Gravano, A.; and Hirschberg, J. 2008. High frequency word entrainment in spoken dialogue. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short '08, 169–172.

Niederhoffer, K. G., and Pennebaker, J. W. 2002. Linguistic style matching in social interaction. *Journal of Language and Social Psychology* 21(4):337–360.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Pennebaker, J. W.; Francis, M. E.; and Booth, R. J. 2001. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates* 71(2001):2001.

Rahimi, Z., and Litman, D. 2018. Weighting model based on group dynamics to measure convergence in multi-party dialogue. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, 385–390.

Rahimi, Z.; Kumar, A.; Litman, D.; Paletz, S.; and Yu, M. 2017. Entrainment in multi-party spoken dialogues at multiple linguistic levels. *Proc. Interspeech 2017* 1696–1700.

Romero, D. M.; Galuba, W.; Asur, S.; and Huberman, B. A. 2011. Influence and passivity in social media. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 18–33. Springer.

Tang, J.; Sun, J.; Wang, C.; and Yang, Z. 2009. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 807–816. ACM.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, 1067–1077. International World Wide Web Conferences Steering Committee.

Wasserman, S., and Faust, K. 1994. *Social network analysis: Methods and applications*, volume 8. Cambridge university press.

Xing, W., and Ghorbani, A. 2004. Weighted pagerank algorithm. In *Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004.*, 305–314. IEEE.

Yu, M., and Litman, D. 2019. Investigating the relationship between multi-party linguistic entrainment, team characteristics and perception of team social outcomes. In *32nd International FLAIRS Conference*.

Zhou, C.; Liu, Y.; Liu, X.; Liu, Z.; and Gao, J. 2017. Scalable graph embedding for asymmetric proximity. In *Thirty-First AAAI Conference on Artificial Intelligence*.