Network Effect on Teams, Team Processes, and Performance

by

Donald Dee Liu

S.B. Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 2022

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Author: Donald Dee Liu
Department of Electrical Engineering and Computer Science
May 12, 2023

Certified by: Ray Reagans
Alfred P. Sloan Professor of Management
Thesis Supervisor

Accepted by: Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Network Effect on Teams, Team Processes, and Performance

by

Donald Dee Liu

Submitted to the Department of Electrical Engineering and Computer Science
on May 12, 2023, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Team communication provides a foundation for the emergence and development of important team processes. This research focuses on indicators of a team's transactive memory processes. Through the use of text analysis and natural language processing (NLP) techniques, we illustrate how teams become more efficient in their work processes and develop a shared problem-solving framework, which in turn are beneficial for team performance. These computational tools allow us to measure and assess the influence of established team processes in an online and distributed work context.

Thesis Supervisor: Ray Reagans
Title: Alfred P. Sloan Professor of Management

Acknowledgments

This research would not have been possible without the wonderful guidance and assistance of a host of people.

My supervisor, Ray Reagans, deserves a great deal of credit for his enthusiasm, his confidence in me, and his terrific advice every step of the way. He's been an amazing advisor in the areas of computational social sciences, online experiments, and text analysis research.

I also offer great thanks to Empirica and all of its contributors, especially Abdullah Almaatouq and Mohammed Alsobay. Empirica has offered a great platform for developing and conducting synchronous and interactive human-participant experiments in a virtual lab environment. I want to thank Abdullah and Mohammed for guiding me through developing and conducting my first online experiment.

I also want to thank my friends and family for their constant support through my undergraduate and graduate years at MIT.

I also want to thank my advisor, Dr. Karen Sollins, for providing me academic advice every semester for the past four years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Unstructured tasks are tasks in which there are no clear instructions or guidelines given to the people asked to complete the task. In addition, these people often have little to no experience with the task itself. Thus, it is largely up to each teammate to learn on the job, plan as they go, and complete the task as they see fit. Some examples of unstructured tasks are increasing water supply for a growing community, designing a new subway system that spans multiple areas, and maximizing the efficiency of a manufacturing processing. In contrast, some examples of structured tasks are converting temperature from Celsius to Fahrenheit, solving for x in an algebraic equation, and other problems that can be solved by following a rule or algorithm. One important characteristic of unstructured task environments is that it is a breeding ground for creativity and collaboration. Unstructured tasks offer the freedom to collaborate more freely with colleagues and provide them with the ability to succeed or fail, which in turn generates a more diverse array of potential solutions.

To establish effective communication and collaboration within an unstructured task, a transactive memory system is often developed. Proposed by Daniel Wegner [23], a transactive memory system (TMS) is a group-level knowledge sharing and memory system in which group members share responsibility for encoding, storing, and retrieving information from different knowledge areas.

Within a team context, a team's shared problem-solving framework defines their TMS (Lewis, 2003 [13]; Ren and Argote, 2011 [18]). A shared problem-solving framework is a communal stock of knowledge that facilitates coordination and learning. It is a memory of their performance or outcomes produced by their past choices and decisions, which provides a foundation for making better choices and decisions (Denrell, Fang, and Levinthal, 2004). The specialized knowledge developed from a TMS allows individuals to learn and develop deep expertise at a faster rate and increases the efficiency of a team. This, in turn, improves team performance. With more time and experience, a shared framework develops and improves.

Teams are said to have developed a shared problem-solving framework or TMS when team members have developed work roles and responsibilities that allow them to create and use specialized knowledge and have learned how to coordinate their behavior. Traditionally, scholars have adopted two distinct approaches to investigate the emergence and effect of a TMS. One approach is indirect and involves measuring the positive association between a team's experience working together and the team's performance. Researchers using this indirect approach suggest that the association between experience working together and team performance can be attributed to the development of TMS. Experience working together has been found to provide team members with opportunities to learn who knows what and develop a division of labor to utilize expertise available on the team most effectively. By doing so, teams learn how to work a collective unit. On the other hand, scholars have also taken a more direct approach. Researchers using the direct approach would sum measures of experience, efficiency in the form of knowledge utilization and coordination to form a measure for a TMS. By doing so, they can analyze the relationship between this composite measure and team performance. (Reagans et al., 2016) [17] Another direct approach involves putting a variety of teams together to work towards a goal and surveying them throughout the process of their work. Surveys would typically ask for their experience working with other team members, what they believe their overall strengths were, what they could do better, etc. Researchers can analyze these survey measures and team performance to measure a TMS.

Although there is merit to the indirect and direct approaches, those methods are difficult to implement within a distributed online context. Teams often leave a residue of their work and communication data after completing their task. This research is an attempt at analyzing the team communication to infer the same indicators of a team's TMS through the use of computational methods such as text analysis and NLP. We will be measuring and investigating two main aspects a team's TMS development: team efficiency and team coordination. With respect to team efficiency, we can observe efficiency through measurements of a team's message rates, message lengths, delays between messages, distribution of messages, etc. With respect to team coordination, we can observe the emergence of a shared language, changes in sentiment, and reliance on direct references to specific team members or to the task to measure if team members have clarity about who does what on the team and about the assignment. In addition, we can take a deeper look into a team's processes by using recent advances in deep learning-based natural language processing (NLP) to inspect the content of the shared language. One potential model is the Discourse Atom Topic Modeling which combines topic modeling and word embedding to detect differences in the content of the communication and what aspect of the team work team members are discussing.

Chapter 2

Related Works

Team communication is integral to a number of team behaviors that lead to effective team performance. Models of teamwork have documented that communication can enhance team performance by facilitating and improving team processes, such as sharing information, clarifying misunderstandings, and providing guidance to other team members. (Marks et al. 2001) [14] In order to analyze communication data, researchers have used a variety of text analysis tools to extract important information and observations from text data.

2.1 Text Analysis

The core of text analysis is training computer software to associate words with specific meanings and to understand the semantic context of unstructured data. This is similar to how humans learn a new language by associating words with objects, actions, and emotions.

Text analysis comes from four research areas: information retrieval, data mining, deep learning, and natural language processing (NLP). Currently, text analysis software works mainly on the principles of deep learning and NLP.

2.1.1 Information retrieval

Information retrieval began in the 1970s. This mainly dealt with text retrieval. Given a query, which can be a few keywords or a full document, we want to find related documents from a text collection or corpus. This has evolved over time into today's web search engines, which are giant information retrieval systems.

2.1.2 Data mining

Traditional data mining uses structured data as database tables. In the late 1990s, researches started to use text as data, which gave rise to text mining. Early text mining basically applied data mining and machine learning algorithms on text data without using NLP techniques.

2.1.3 Deep learning

Artificial intelligence is the field of data science that teaches computers to think like humans. Machine learning is a technique within artificial intelligence that uses specific methods to teach or train computers. Deep learning is a highly specialized machine learning method that uses neural networks or software structures that mimic the human brain. Deep learning technology powers text analysis software so these networks can read text in a similar way to the human brain.

2.1.4 Natural language processing (NLP)

NLP is a branch of artificial intelligence that gives computers the ability to automatically derive meaning from natural, human-created text. It uses linguistic models and statistics to train the deep learning technology to process and analyze text data, including handwritten text images.

2.2 Latent Semantic Analysis and Cosine Similarity

Early attempts at moving towards digital analysis of team communication data used latent semantic analysis (LSA). LSA is a statistical computational method that decomposes documents to a vector representation of their semantic meaning by applying singular value decomposition on a matrix of word frequencies by documents. These LSA vector representations can be used to find the similarity of two documents by taking the cosine distance of their respective semantic vectors. Researchers have found success in using the outputs of LSA to examine team communication content (Foltz et al. 2006) [10], detecting patterns of communication (Foltz et al. 2009) [9], and analyzing team cognition (Gorman et al. 2013) [11]. However, this approach suffers from several limitations. LSA is unable to account for a number of linguistic features which impact the quality of its semantic representations, such as polysomy, word ordering, and syntactic structure. Furthermore, the cosine method does not easily incorporate hierarchical representations of conversations (e.g. who is speaking, what task is being performed, etc) and is bounded by the length of the documents being compared (Spain et al. 2019) [21].

2.3 Word Embedding and Word2Vec

Another important technique in text analysis was word embeddings. Word embeddings is a technique where individual words are transformed into a numerical representation of the word (a vector). The vectors try to capture various characteristics of that word with regard to the overall text. These characteristics can include the semantic relationship of the word, definitions, context, etc. With these numerical representations, you can do many things like identify similarity or dissimilarity between words. However, there are multiple limitations of simple embeddings such as this, as they do not capture characteristics of the word, and they can be quite large depending on the size of the corpus.

In 2013, Mikolov et al. [15] introduced a popular word embedding model called

Word2Vec. Although vector representations of words have been used as early as 2003 (Bengio et al.) [4], the main innovation proposed in their paper was an efficient improvement of the training procedure, by removing the hidden layer and approximating the objective. Together with the efficient model implementation, these simple changes enabled large-scale training of word embeddings on huge corpora of unstructured text. Given a large enough dataset, Word2Vec can make strong estimates about a word's meaning based on their occurrences in the text. These estimates yield word associations with other words in the corpus. For example, words like "King" and "Queen" would be very similar to one another. When conducting algebraic operations on word embeddings you can find a close approximation of word similarities. Word2Vec algorithms aim to give equal weights to all words in the corpus. However, from a contextual perspective, some words are more important than others. Furthermore, it doesn't have a good technique to deal with ambiguity. Two exact words but in two different contexts will have too close vectors.

2.4 Discourse Atom Topic Modeling

To better summarize large corpora and find evidence for patterns in team communication, we can utilize discourse atom topic modeling (DATM). DATM integrates topic modeling and word embedding to capitalize on their distinct capabilities. Topic modeling methods identify latent themes in a corpus and connect those themes to observe words and documents. Word-embedding methods represent word meanings by mapping each word in the vocabulary to a point in an N-dimensional semantic space. Words used in similar contexts in the corpus are mapped to nearby points. Like a standard topic model, DATM identifies topics or latent themes and infers the distribution of topics in a specific document. However, DATM does so in an explicit embedding framework such that both words and topics live in one semantic space. DATM offers rich representation of topics, words, phrases, and latent semantic dimensions in language (Foster et al. 2022) [2].

2.5 Modern neural language models

Modern neural language models such as fastText (Bojanowski et al. 2016) [5] and BERT (Devlin et al. 2018) [8] have been observed to perform better on a range of NLP tasks because they are less sensitive to corpora training when compared to LSA. fastText associates embeddings with character-based n-grams and attempts to capture morphological information to induce word embeddings and deals better with out of vocabulary words. BERT makes use of a bidirectional Transformer to learn contextual relations between words in a text. (Vu et al. 2019) [22]

Chapter 3

Experiment Design

3.1 Experiment Design

The team communication data will be collected through renovating the Bavelas-Leavitt-Smith classic team coordination experiment which focused on estimating network effects on team performance.(Leavitt, 1951) [12]. (Burt et al., 2021) [6] designed the renovated experiment. Participants from crowdsourcing platforms like Amazon Mechanical Turk and Prolific are assigned at random to positions in five-person teams. Within each team, a network of communication channels is defined by restricting communication access between teammates. We used a total of four different networks, shown in Fig 3-1.

The baseline is a Fully Connected or Clique network in which every teammate can communicate with every other teammate. Of the 10 communication channels in the network, every teammate is involved in four (40% of team communication is concentrated in any one teammate). The opposite extreme is a Wheel network in which a team leader coordinates the activities of all four teammates. 100% of communication is concentrated to the player at the center of the Wheel. The other two networks are less centralized variations of the Wheel. The Disconnected Brokers (DB) network has two leaders (brokers) independently coordinating the activities of three shared subordinates. These leaders are both involved in 50% of the team's communication. The Connected Brokers (CB) network is a DB network, but the two

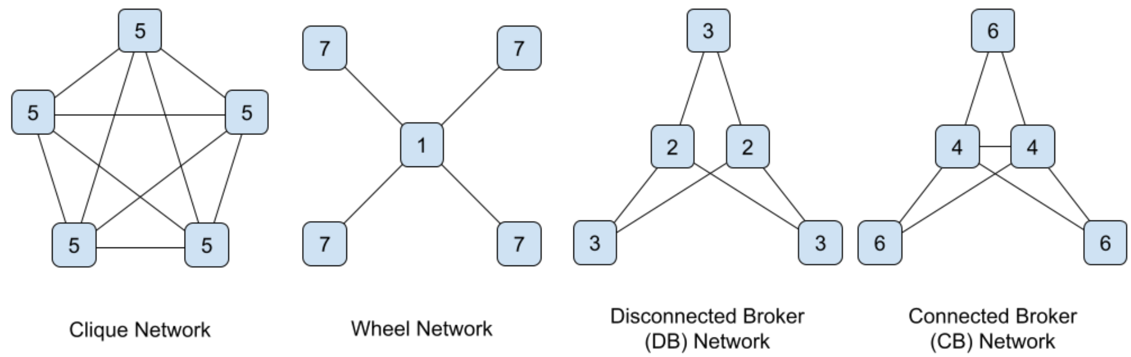

Figure 3-1: Communication Network Structures. In a clique or fully connected network, all players are connected to all other players. On the other hand, in a wheel network, all players are connected to only the central player. A DB network is a variation of the wheel, in which there are 2 leaders each connected to three other players. A CB network is a DB network, but the 2 leaders are connected to each other.

leaders can communicate directly with each other. Within the CB network, the two leaders are involved in 57% of the team's communication. The CB network can also be seen as three overlapping Cliques, each containing the two leaders plus one shared subordinate. For reasons of a division of labor in which groups specialize on separate bits of work, or variation due to the independent evolution of separate social groups, we manipulated the network structures of the teams. (Burt et al., 2021) [20]

Subjects can only communicate with teammates they are allowed to communicate with through one-on-one chat boxes. This means they do not see messages between their teammates. Furthermore, they do not see the communication network structure.

Each team is asked to play a total of 60 minutes, up to a maximum of 15 trials. In each trial, each person in the five-player team receives a set of five symbols, with one symbol common to the sets of all five teammates. The task of each trial is to identify the shared symbol by communicating with teammates in their communication network. A subject can submit his or her best guess of the common symbol. Subjects know how many teammates have submitted answers, but do not know which teammates have submitted, nor what they submitted (unless specified by the teammates through sending a personal message). When all five subjects have submitted their best guess, the trial is over. Subjects are allowed to reconsider their answer after they submit and before all five subjects submit if they receive new information from their teammates or decide to change their mind.

The superset of symbols are made up of tangram symbols taken from a well-known study of language coordination (Clark and Wilkes-Gibbs, 1986) [7]. These symbols are shown in Fig 3-2. To do well, team members must describe the tangrams using similar words and phrases. In other words, they must develop a shared language to successfully coordinate and identify the common symbol.

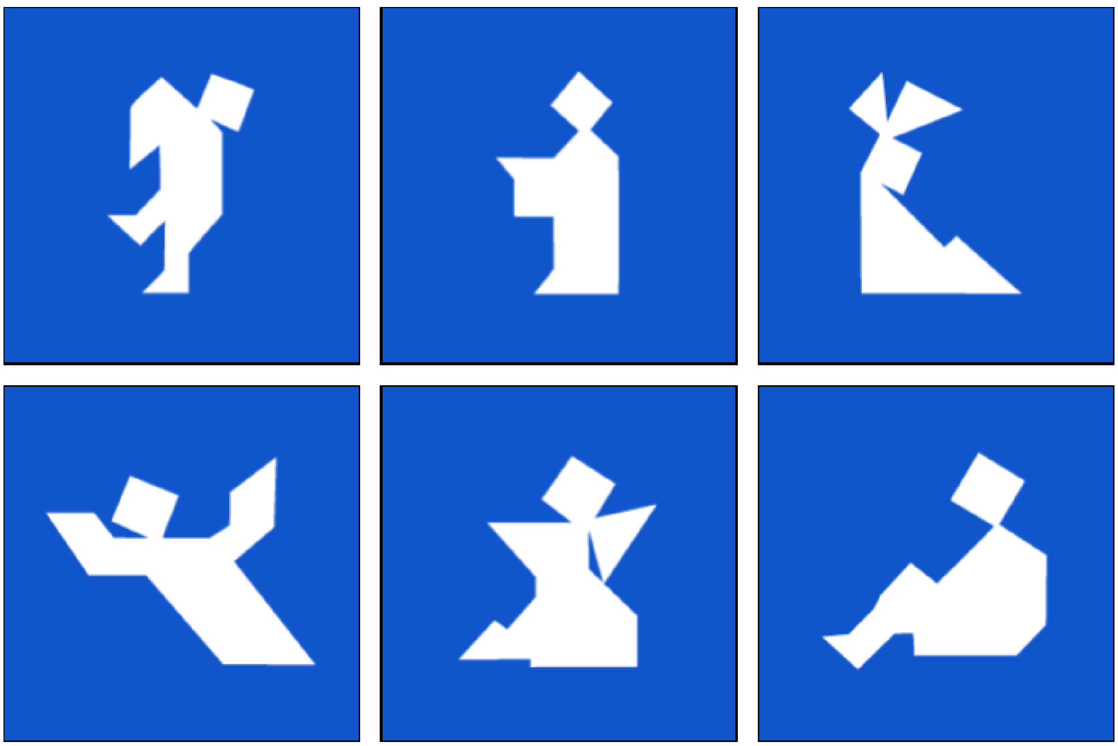

Figure 3-2: Superset of Tangram Symbols. The following six tangrams were used to create 6 distinct card sets of 5 cards, each card set missing one of the six symbols. Each trial, 5 card sets will be distributed amongst 5 players. Only one of the symbols among all of the distributed card sets will be shared.

Subjects receive a fixed payment plus an additional incentive payment for each trial in which all five subjects correctly identify their shared symbol. The incentive payment encourages players to play more rounds and put effort into each round they play. We study the messages within trials a team completed. Message content and

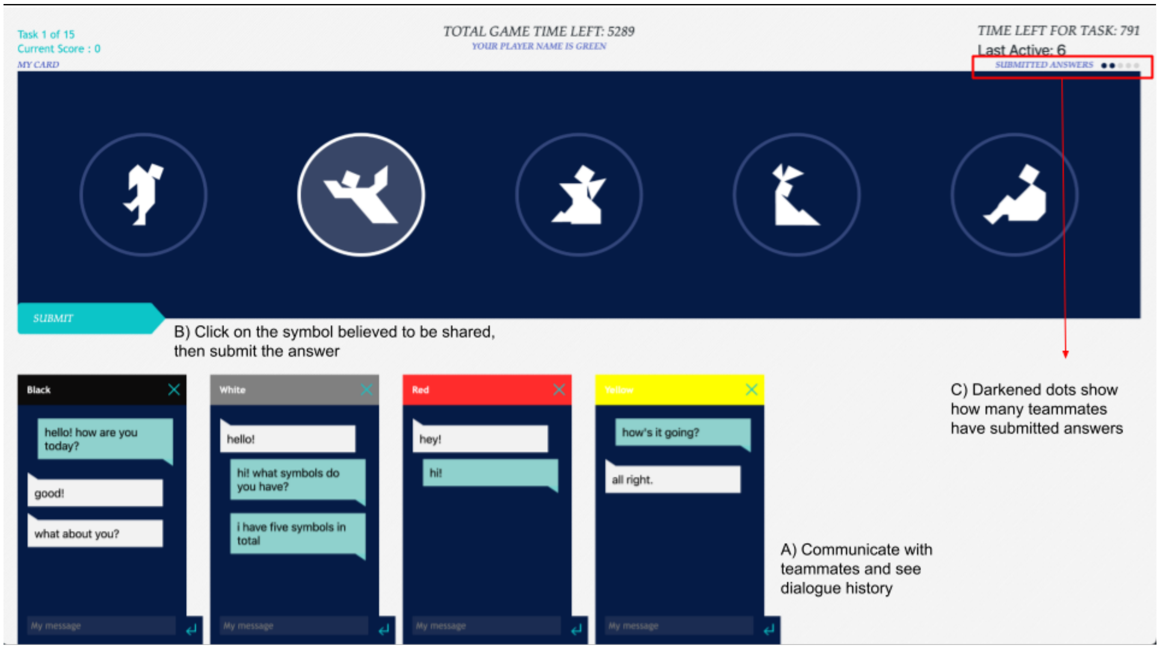

Figure 3-3: Example of experiment game screen. Subjects can communicate with team members in their communication networks through individual chats to discuss which tangram symbols they share. (A) They can select the symbols in the My Card Section and submit their answer if they believe it's the shared symbol. (B) In the top right corner, you can see the number of teammates that have submitted their answers by counting the number of darkened dots. (C)

timing are recorded.

3.2 Preprocessing Data

Before working with the communication data, I first preprocessed the data to simplify the text for the classifier to learn the features easily. First, I removed unicode text from raw text to overcome Python unicode decode error. Then, I performed emoticon and punctuation tagging. Emoticons and several punctuation marks can be converted into indicators or tags as these symbols can provide insight into an individual's emotional state. Other punctuation marks that are not relevant will be removed. I also expanded chat abbreviations. Within chat language, many abbreviations in the form of short forms or acronyms are often used and need to be expanded. Short forms are short representations of a word created by omitting or replacing a few characters e.g.

lol $\rightarrow$ laughing out loud, w $\rightarrow$ with, etc. It is to be noted that all chat abbreviations cannot be replaced by a simple expansion, as some meanings vary in different contexts, e.g. "btw" can refer to "by the way" or "between". I also expanded any English contractions. In addition, I performed spell checking on all word tokens by using a package called SpellChecker. Finally, I performed a basic named entity recognition by mapping a player's anonymous color name (e.g. Purple, Blue, Yellow) into a person category and any converged symbol jargon into a symbol category.

Chapter 4

Team Efficiency and Coordination

Within our experiment, one way to observe a team's shared problem-solving framework or TMS is to measure a team's efficiency. The specialized knowledge developed from a TMS allows individuals to learn from their decisions from past trials and increases their efficiency as subsequent trials. If a team gets the shared symbol wrong, players can clarify any misunderstanding about the symbols they picked, discuss the team's decision-making system, etc. If a team successfully passes a trial, they can reflect on the process that resulted in them all picking the right symbol and build upon their collective communication skills. As teams play, they should become more efficient in sharing meaningful information during each trial and organizing that information to find their shared symbol. This in turn, improves team performance.

Within research on effective communication patterns, researchers have identified three aspects of effective communication that affect team performance (Pentland, 2012) [16]. The first is efficiency, which is measured by the number and nature of exchanges among team members. Team efficiency revolves around the volume and frequency of messages. The second is coordination, which reflects the distribution of energy among team members and the organization of players and information. If all members of a team have relatively equal and high engagement and understanding with all other members, coordination is strong. The third is exploration which involves communication that members engage in outside their team. Exploration is often controlled by the structural holes within a team's communication networks.

Although exploration is less feasible in our experiment environment, team efficiency and coordination can be easily explored by measuring the number of exchanges and the structure of each team.

4.1 Team Efficiency

To quantify the efficiency of each team, we can observe patterns in a team's communication volume and frequency. We expect that as subsequent trials pass, teams would send less words and messages at a faster rate. If teams develop a shared problem solving framework, players can better communicate concise, yet meaningful information to help them succeed in successive trials. By analyzing a team's communication volume and frequency patterns such as their words per message, messages and words sent per trial, number of messages and words sent over the duration of a trial, and the average lag between pairs of individuals and consecutive messages.

4.1.1 Average Number of Words Per Messages

As teams go through subsequent trials of the game, Fig 4-1 shows that the teams are sending fewer number of words per message: an average of 7.5 words in the first three trials, down to 4.5 words in the last three trials. We expect the average number of words to plateau around 5 words, correlating with the number of symbols a player sees on their screen at a given time. We can see that teams are improving their coordination as they complete the task with less words overall. They learn to communicate more effectively as they learn from their past trials what words they should focus on sending to give players the significant information they need. This curve is statistically significant and robust to different fit statistics. The network structure does not have a significant effect on the average number of words per messages as all teams exhibit similar quantities in the number of words per message and are showing the same increase in efficiency.

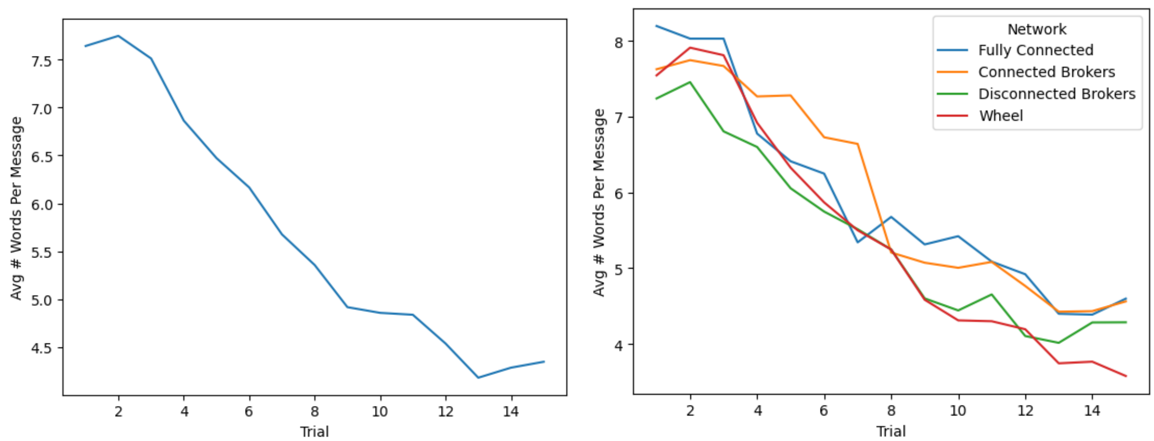

Figure 4-1: Average number of words per message. The number of words within each message decreases per trial.

4.1.2 Average Number of Words and Messages Per Trial

As teams go through subsequent trials of the game, Fig 4-2 shows that teams are also sending fewer messages per trial: with over 100 messages sent in the initial trials, down to 40 messages in the final trials. Teams are not only sending less words per message, but also less messages overall to complete the same tasks. This shows that teams are becoming more efficient in their communication by filtering out even more information and understanding what information each person needs to send.

Similarly, in alignment with the teams sending less words per messages in later trials, Fig 4-3 shows that teams are also sending less words per trial: with over 700 words in the initial trials, down to less than 200 words in the final trials. Again, this indicates an improvement in effective communication.

It is important to note that the Wheel network is an outlier in this measurement. The Wheel network has less channels for communication, so there are less opportunities for messages to be sent. For the remaining network structures, the DB network, on average, has the largest number of messages sent per trial. The CB network, similar to the DB network, starts off with one of the higher number of messages per trial, but quickly drops. As subjects learn their role in the network structure and the two brokers are able to directly share their information between the two independent

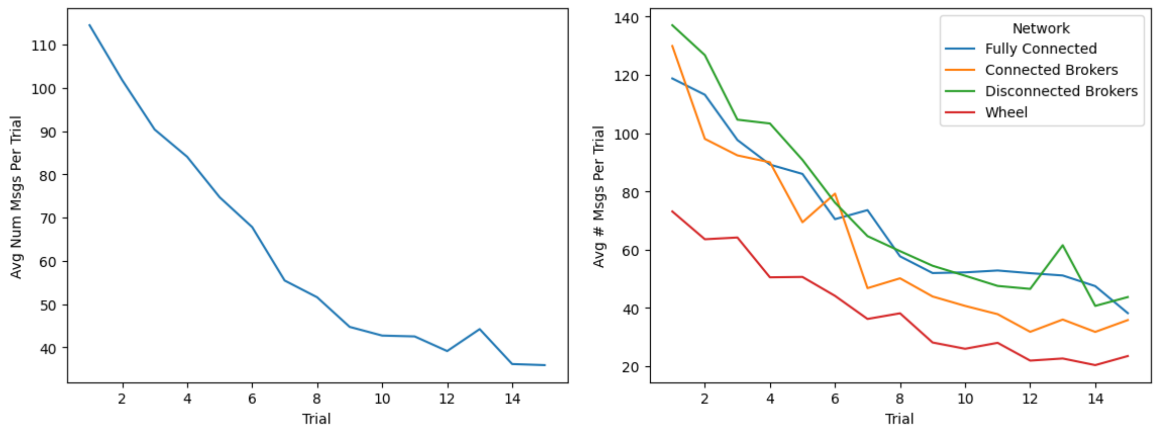

Figure 4-2: Average number of messages per trial. The average number of messages per trial decreases.

3-person cliques, the sharing of information becomes more concise and efficient. In contrast, the DB network has no direct way to connect information between the two independent cliques and requires the ends of the cliques to serve as a middleman in the information sharing process. This increases the number of messages and decreases the efficiency at which information is collected. Regardless, the trend that the number of messages or words sent decreases as subsequent trials pass holds for all networks. This shows that these networks are developing a more efficient communication system.

4.1.3 Average Number of Words and Messages Per Duration of Trial

Another aspect to examine a team's communication efficiency through subsequent trials is to observe the number of messages and words sent per duration of a trial. Fig 4-4 shows that the rate at which messages are sent throughout a trial increases as subsequent trials pass: from less than 0.25 messages sent per second (15 messages per minute) in a trial up to 0.40 messages sent per second (24 messages per minute) in a trial. Teams are developing the messages they need to send at a faster rate. They learn the process of telling their teammates the significant and important information each round and iteratively refine the process with each consecutive round. This trend

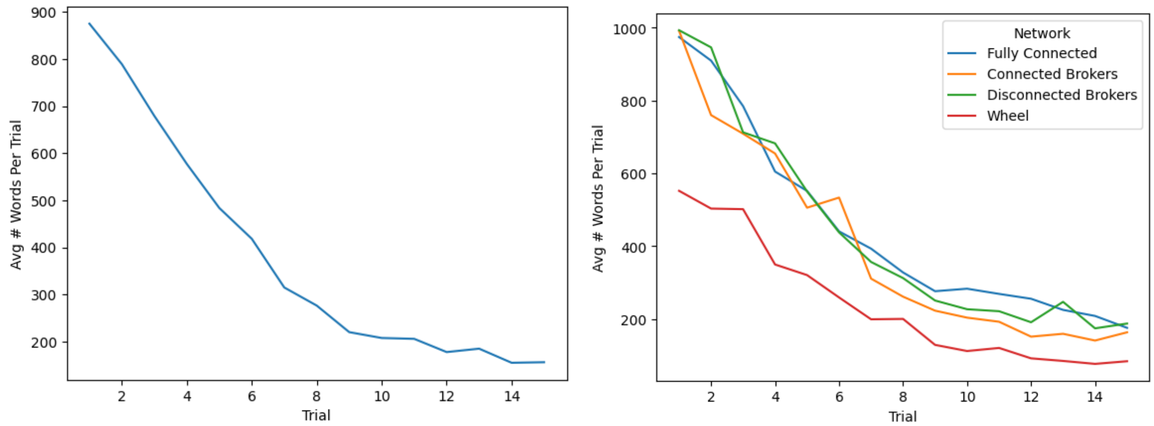

Figure 4-3: Average number of words per trial. The average number of words per trial decreases.

is in line with less words being sent per message. With less words being typed, more messages can be sent within the same period of time.

An interesting observation was that average number of words per duration of a trial hovered around the same value throughout the trials. This is a consequence of the increase in the average number of messages sent per round and the decrease in the average number of words sent per message. Therefore, even with the same number of words being sent throughout a trial, a team can increase their performance by adjusting and refining the words being sent to be more informative and meaningful for the task at hand. Although we do not see an increase in efficiency with the average number of words sent per duration of trial, we can suspect that the nature and content of these words are shifting, which will be analyzed in the later sections.

It is again important to note that the Wheel network is an outlier in this measurement, because the Wheel has less communication channels, which reduces the overall number of message being sent. However, the trend that the rate of messages being sent per trial increases and that the rate at which words are being sent per trial stays constant as successive trials pass holds true. The effectiveness of each team's TMS is improving.

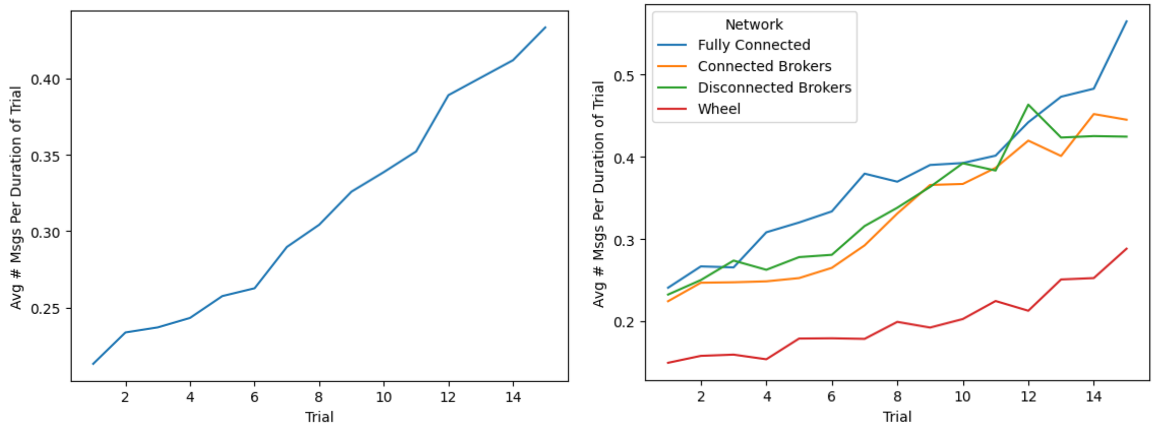

Figure 4-4: Average number of messages per duration of trial. The rate at which messages are sent per trial increases.

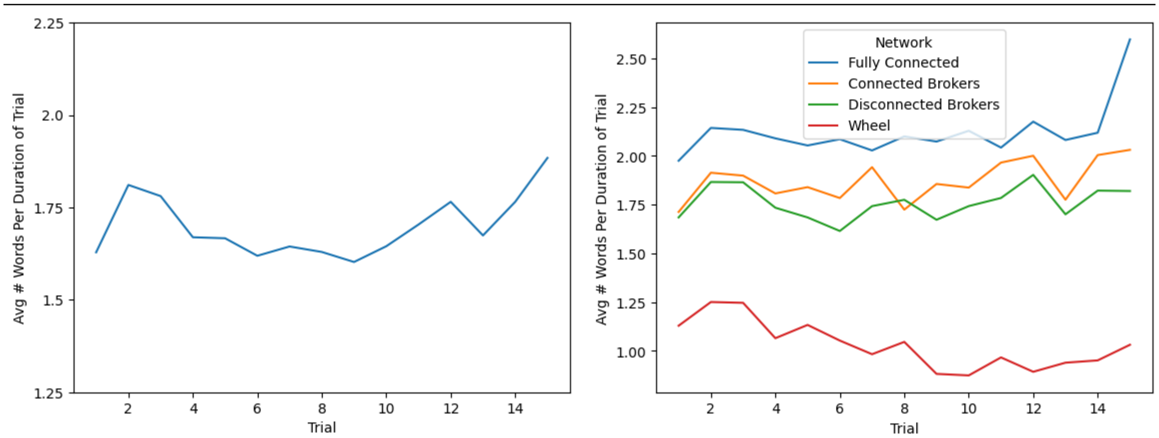

Figure 4-5: Average number of words per duration of trial. The rate at which words are sent per trial stays constant. This is an effect of a higher rate of messages sent, but less words per message.

4.1.4 Average Time Delay Between Pairs

An additional method to observe the rate at which messages are being sent is to compare the time delay of messages sent between the pairs of players within a network. Fig 4-6 show that as subsequent trials pass, the time delay between messages between a pair of players decrease: from more than 60 seconds between messages between pairs, down to less than 35 seconds between messages between pairs. Players are more efficient with their messages and are able to message different players quicker. Players, over time, learn to know what kind of messages or information to expect from other players as well as know what others expect from them. This allows them to transmit significant information faster and in a more organized fashion to other players.

An interesting trend to note is the ranking of time delays between the different networks. From the longest time delay to the shortest, the networks rank Fully Connected(FC), Connected Brokers (CB), Wheel, Disconnected Brokers (DB). Our initial expectations were that the Wheel would have the longest time delays between the central player and the players at the edge of the wheel, since the players at the edge can only speak to one person or wait for a response. However, the FC and CB have the longest time delays between. We believe that each player has more communication channels they can be talking through and it takes players longer to formulate a response/answer or comprehend all the information delivered to them. The Wheel network ranks third in terms of time delays. Lastly, the DB network has the shortest time delays between pairs. Without a central leader paired with a low number of connections per player, the DB network has less of a need to coordinate between everyone and has less information to sort through and can therefore communicate quicker.

4.1.5 Average Time Delay Between Messages

Furthermore, we can look at the time delay of consecutive messages sent by a player within a network. Fig 4-7 shows that as subsequent trials pass, the time delay be-

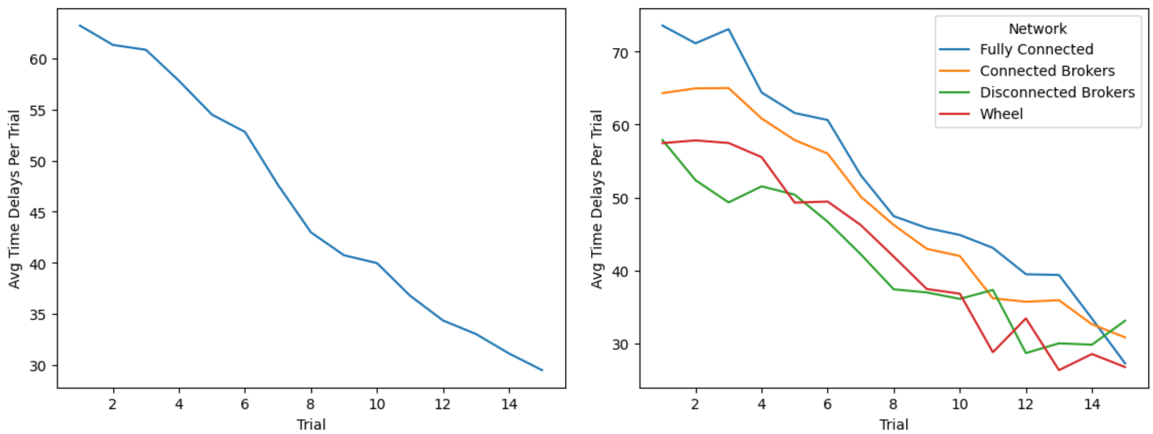

Figure 4-6: Average time delays between pairs per trial. The rate at which players respond to the same teammate within a team decreases per trial.

tween consecutive messages from a player also decrease: from 28 seconds between consecutive messages down to less than 14 seconds. Players become more efficient at sending messages as they are sending consecutive messages faster. They understand the type of information they need to transmit to their teammates and also the type of response that their teammates need to coordinate all of the information to submit an answer. Along with the general effectiveness of their message lengths, they not only exchange messages faster, but they also exchange more meaningful messages.

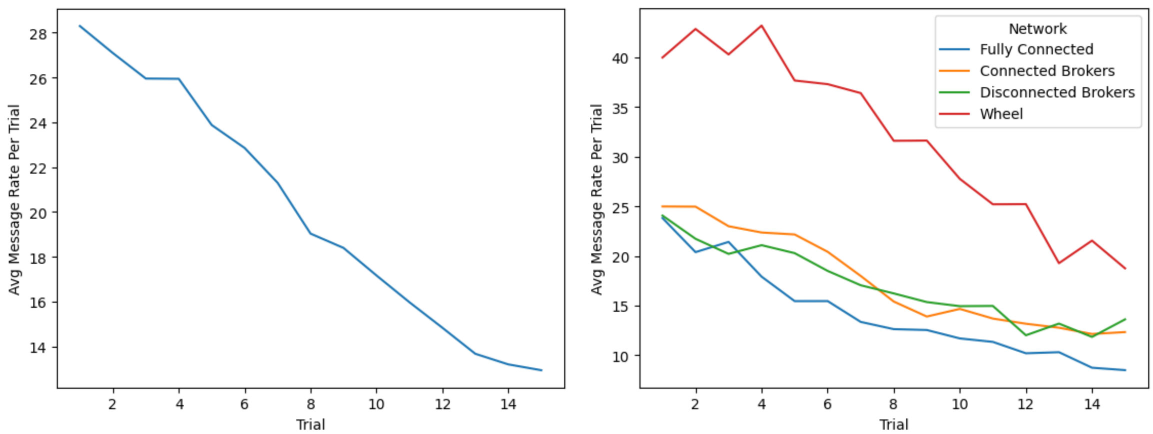

Figure 4-7: Average time delays between consecutive messages per trial. The rate at which players send consecutive messages (to any teammate) decreases per trial.

Chapter 5

Team Coordination

Another important aspect of a shared problem-solving framework or TMS is to observe team coordination and learning through examining patterns within the communication data. Within communication data, we can observe the development of a shared language, sentiments towards specific players or task, and organization of communication.

5.1 Shared Language

5.1.1 Consecutive Message Similarity Per Trial

One important aspect in a team process is how a team shares information with each other. Within this unstructured task, we saw the emergence of a shared language in which teams learned to coordinate on using the same labels and descriptions for different symbols and structure their information in a similar syntax. A shared language is also a form of a TMS (Moreland et al., 1996)[19]. The path of a shared language measurement compared to team performance reflects the importance of a group or team developing a shared problem-solving framework.

To quantify the shared language, we graphed the average similarity between each consecutive message within a trial. Fig 5-1 shows that subsequent message within a trial tend to become more similar as trials pass, indicating the emergence of a shared language. The message vectors being sent by different players are closer in terms of vector similarity because teams develop their own vocabulary for the different symbols and standardize the way they send information to each other (i.e. listing the symbols they see from left to right, listing symbols one at a time, etc.) With the the development of a TMS through a shared language, players are able to effectively coordinate the information they're transmitting by using terms that the entire team understands. By having all of the players on the same page, teams can clarify any misunderstandings and understand what each member is trying to convey. This shared language is a form of tacit coordination in which team members can understand each other without any formal agreement. This shared language correlates with higher performance.

The network structure seems to have a significant effect on a team's shared language. Networks that have more communication channels like the Clique and CB networks seem to develop their shared language faster and more effectively. Networks that have less communication channels like the DB and Wheel networks seem to have a harder time developing a shared language. The disconnect between the two independent 3-player cliques within the DB network make it difficult for all 5 players to coordinate on a standardized way of communicating. The Wheel network does not seem to successfully develop a shared language. The Wheel network may reason that they do not need to develop a shared language if the single player in the central hub simply dictates what the other players should do after receiving information from everyone. The central player organizing everyone's information and delegating what players should do is a form of explicit coordination. Whether it be tacit or explicit coordination, teams develop their own shared problem-solving framework in order to coordinate their symbol information to find the shared symbol.

5.2 Direct References In Messages

Another important characteristic within the communication data that we observed was the references to specific subjects, specifically towards different players and the

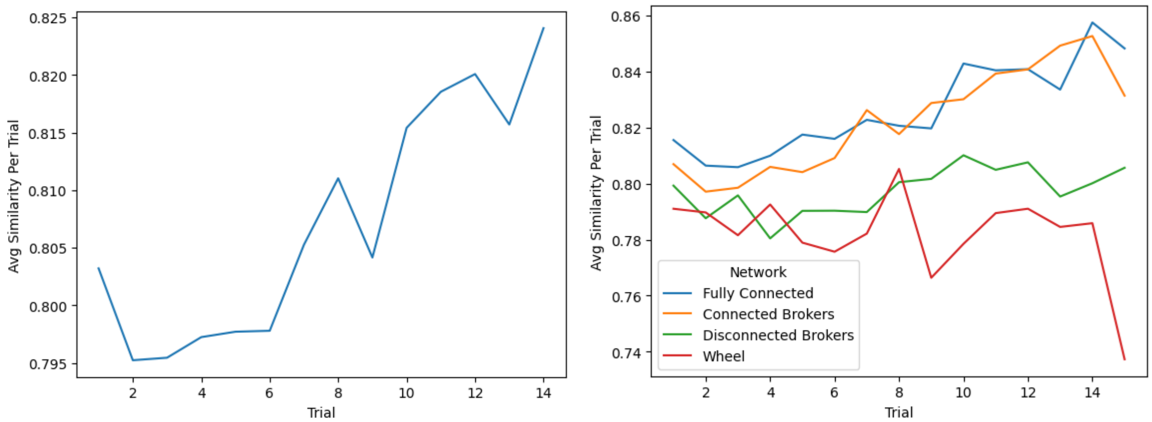

Figure 5-1: Average similarity of messages per trial. The average similarity score between a team's cumulative messages within a trial (sorted by timestamps) increases per trial.

overall game itself. References to players or the game also offers insight into what aspects of the TMS are being developed or improved upon. Specifically, messages containing these direct references provide details on work and role clarity and coordinating player-relations within the game.

To do this, we first had to do entity tagging. Players could be tagged by certain key identifiers (i.e. their anonymous color names). Tagging the overall concept of the game or research required training a Word2Vec model using the player messages as the input corpus. Then, we looked for any words that had a similar word vector to the words "game" and "research". We examined the similar words within the context they were used to verify that they were referencing the game. With the following entity tagging, we extracted all messages that directly referenced the players and the game.

5.2.1 Average Number of Player and Game References

As the game progressed, we calculated the average number of player and game references that were being made.

Fig 5-2 show that the average number of player references across messages tend to decrease as trials pass. The need to reference players is a tool for coordinating information and answers between players. As a team develops their TMS, we can see the content of the messages directly reference players less. In the beginning of the game, players need to discern which role they play within their communication network. Their role determines what type of information they need to send, what type of information they'll receive, and what they should do with that information. To do so, players will often reference specific players to clarify what each player is doing. Over time, the team will have enough experience to expect what each player will do and coordinate their actions to theirs. For example, at the beginning of a new trial, players at the ends of a wheel will send information to the central player. The central player can then organize all of the information and tell each player what everyone has in common. Although the team coordination process may not be that simple, teams will still find their own system to coordinate everyone's roles. As they gather more experience, they will also need to reference players less, since they can expect the players to behave in a controlled fashion. Players begin to understand their role within the team structure and are able to request or send the information without players directly calling out specific players for it.

The structure of the network has a significant effect on the use of player references. For the DB and CB networks, there is a need to coordinate answers between different parts of the structure. Since, they are not all connected to each other, they need to make extra sure that their answer matches up with someone they can't speak to. Within the FC and Wheel networks, there is less of a need to refer to players. For FC networks, every player can receive the information they need to figure out the shared symbol amongst themselves. For Wheel networks, everyone can only talk to the central player and the central player often organizes all of the information and dictates the other players what to choose. This process has little need for referencing individual players. Overall, communication networks that only allowed for indirect communication saw more references to players while communication networks with direct communication saw fewer references to players.

Fig 5-3 show that the average number of game references across messages tend to decrease as trials pass. We expect there to be more references to the game in the initial trials as players get accustomed to game instructions and game objectives. In the beginning of the game, some players may be confused about the game structure or rules. They may ask other players what the goal of the game is or how they should be interacting with the game interface. As players play a few rounds and develop their problem-solving framework through trial and error, they become familiar with the flow of the game and are more coordinated in understanding the goals within the team and the game. Thus, they find clarity within their work and begin to refer to the game. An exception that we saw within this trend was on the final trial of the game. Players suddenly refer to the game more. We suspect that players are offering their final thoughts on the game, such as whether or not they enjoyed it or if they achieved the goals they set at the beginning.

The only significant effect that the networks had on the average number of game references were in the Wheel network. Non-central players were often confused on why they could only speak to one player and raised concerns that there were mistakes in the game. Other than that, there were not any major effects that the networks had on the average number of game references. It is to be noted that there were not as many direct game or research references, so the trends are less robust than the player references.

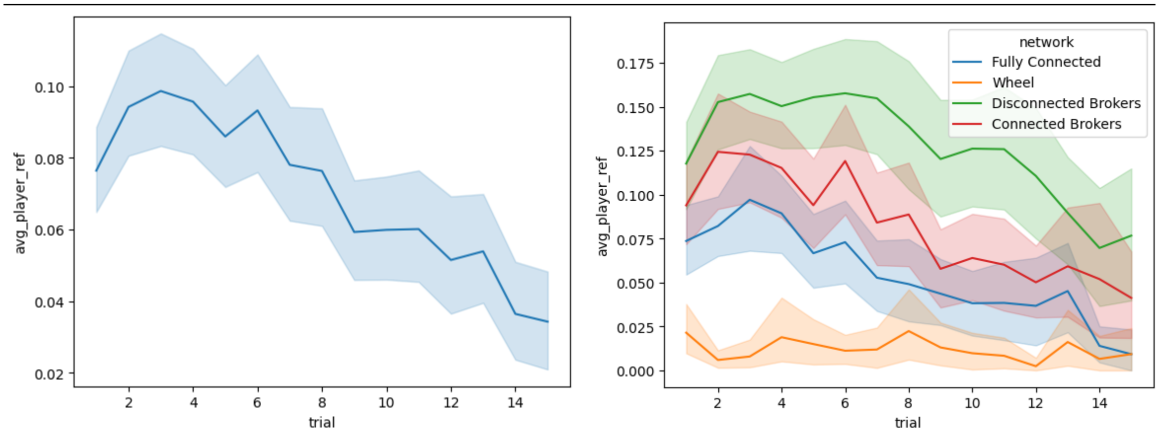

Figure 5-2: Number of player references per trial. The number of player references per trial decreases.

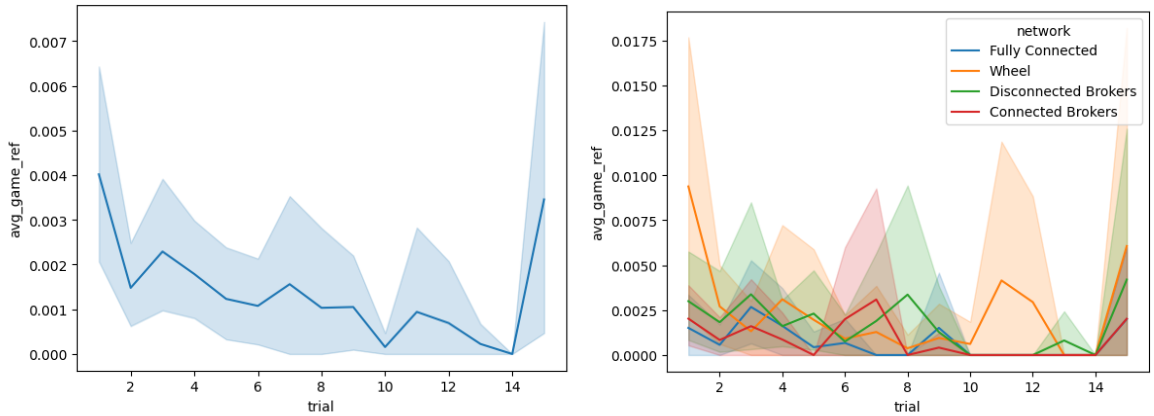

Figure 5-3: Number of game references per trial. The number of game references per trial decreases.

5.3 Shared Sentiments

Another area where we can investigate team coordination is by looking at the shared emotions within a team. A team's mental state can offer important insight into how a team behaves and performs. We expect that teams with a positive mindset will perform better than teams with a negative mindset. Positive emotions build important skills and relationships crucial to optimize performance. By measuring a team's emotional state, we can understand how a team develops a shared emotional framework and how it affects the team's work roles and work.

To measure a team's emotional state, we perform sentiment analysis. Sentiment analysis measures the quality of the team communication. Team communication quality can be referred as the extent to which the communication is perceived as informative, significant, and helpful for the task at hand. In addition, we can refer to communication quality as whether the experience of the interaction was evaluated as positive and enjoyable. Traditional methods to measure communication quality often involves cross-sectional surveys in which team members are asked to rate the extent to which team members share their knowledge or information is openly shared among team members using rating scales. Historically, the process to analyze team communication has been resource-intensive due to the need of researchers to manually examine the surveys and interactions to see if they were positive or not. Furthermore, analysis often only happens post-experiment. This means players are not able to receive feedback on the quality of their communications in real-time to help improve their team performance.

To bring light to recent advances in NLP, we wish to introduce the potential of sentiment analysis, specifically polarity and subjectivity values, into team communication quality analysis.

Sentiment analysis is the process of determining the attitude or the emotion of the writer, i.e., whether it is positive or negative or neutral. Sentiment analysis can also offer insight into a player's energy and engagement level. For example, happy team members may be more excited to participate and willing to engage with more people whereas frustrated team members may become unresponsive and give responses unrelated to the task. To measure a player's sentiment, we utilized the sentiment function of TextBlob which returns two properties, polarity and subjectivity.

Polarity is a float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement. Subjectivity is also a float which lies in the range of [0,1]. Subjective sentences generally refer to personal opinion, emotion or judgment whereas objective refers to factual information.

5.3.1 Polarity

As the game progressed, we calculated the average cumulative polarity score across a trial as well as the average polarity score per message. We graphed this against the proportion of players that submitted the correct symbol per round.

Fig 5-4 shows that the total polarity as subsequent trials pass decreases and hovers around a neutral polarity. Our initial hypothesis was that a low polarity score correlates to less players getting it correct and a high polarity score correlates to more players getting it correct. However, Fig 5-4 shows that teams start off with an overall positive polarity, most likely from initial greetings, casual talk, and ice breakers. Then, the teams tend to become more neutral. We expect that as the team becomes accustomed to the flow of the game and develop their TMS, players become more efficient and systematic in their messages, only sending what's necessary for the round to be successful. This necessary information revolve around words or labels describing the symbols, which have a neutral polarity score since they are mostly simple nouns and verbs with no emotions attached. It is important to note that Fig 5-4 does not indicate that a positive polarity score results in a lower correct percentage. However, it does indicate that as teams perform better, their messages have less positive emotion attached to them. In other words, as a team's TMS develops, players will be more focused on the task and put aside emotions that may disrupt the team processes. There did not seem to be a significant effect of the network on the overall polarity of a team. They all had similar trends of gravitating towards a neutral polarity score as subsequent trials passed.

Fig 5-5 shows the average polarity across a message in each trial. Similarly, as subsequent trials pass, the average polarity score of each message decreases. As a team develops their shared problem-solving framework, there is less of a need for messages that carry emotional weight. Players understand that the task requires factual and straightforward communication to coordinate which symbol is shared amongst everyone. Emotional content will only disrupt the team and not contribute to the overall goal. Thus, teams work to be more coordinated in terms of their goals and mindset to get as many correct as possible. Again, there is no significant effect of the network on the overall polarity of each message. They all had similar trends of messages that neutralized as the game progressed. There were some outliers in the final rounds in the Wheel network that caused the graph to be less linear. If they were excluded, the linear graph would be more defined.

5.3.2 Subjectivity

As the game progressed, we also calculated the average cumulative subjectivity score across a trial as well as the average subjectivity score per message.

Fig 5-6 shows that teams start off with an overall high subjectivity score which gradually neutralizes as more trials occur. We suspect that early trials includes subjective messages including greetings, strategy opinions, initial thoughts on the game

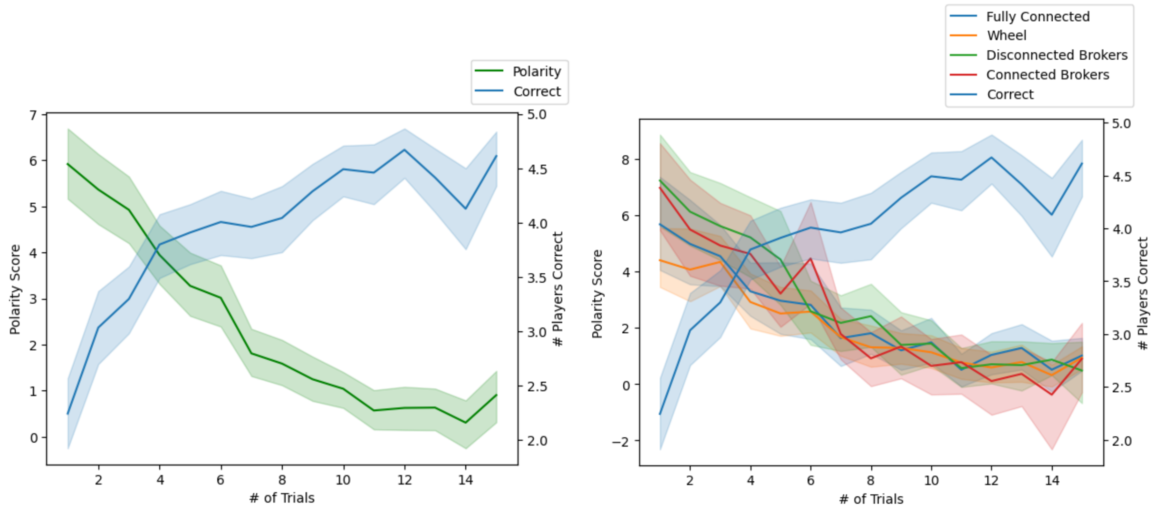

Figure 5-4: Cumulative polarity score vs Correct proportion. The cumulative polarity score within a trial decreases per trial as the proportion of players selecting the correct shared symbol increases per trial.

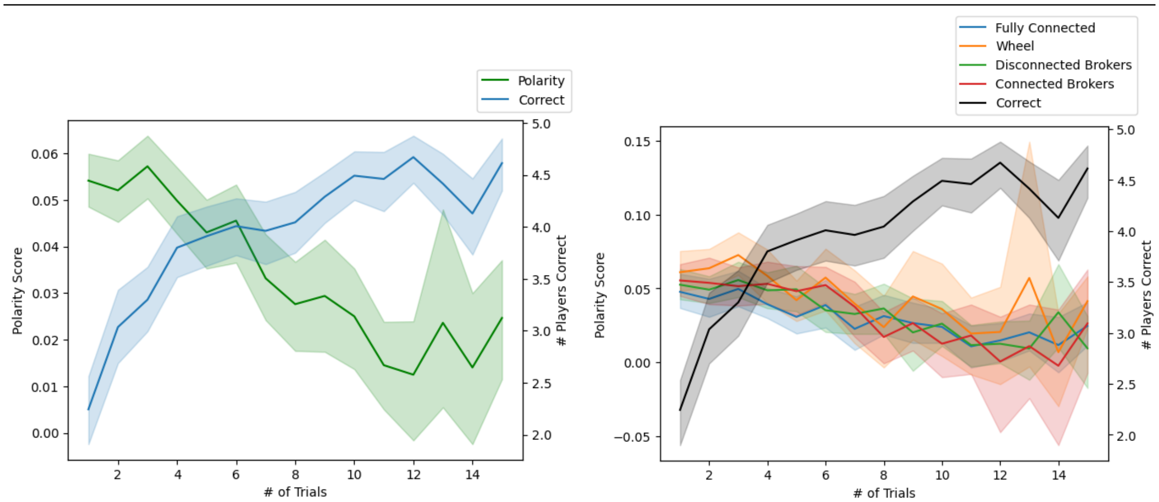

Figure 5-5: Polarity score per message vs Correct proportion. The average polarity score per message decreases per trial as the proportion of players selecting the correct shared symbol increases per trial.

and players. As the game progresses, there is less of a need for opinionated statements as each player understands the overall strategy within the shared problem-solving framework. In the beginning of the game, we expect that a lot of players will provide their personal input on what they think symbols look like, how they think the game should be played, and what they think the best strategy is. Over time, teams will settle on a team strategy and players will no longer send subjective messages on what they believe the team should do. Thus, the decrease in subjectivity is another identifier for an improvement in team coordination. Again, it is important to note that Fig 5-6 does not indicate that a positive subjectivity score results in a lower correct percentage. However, it does indicate that as teams perform better, their messages have less emotion and opinions attached to them. Furthermore, it implies that as teams develop their TMS by understanding their roles and coordinating information, the overall subjectiveness of the team neutralizes. There did not seem to be a significant effect of the network on the overall subjectivity of a team. They all had similar trends of gravitating towards a neutral subjectivity score as subsequent trials passed.

Fig 5-7 shows the average subjectivity across a message in each trial. In alignment with the overall subjectivity, as subsequent trials pass, the average subjectivity score of each message decreases. Within the earlier trials, players will be unsure of how to describe their symbols so that others can understand and speculate what symbols they do or do not have in common with others. As their TMS develops, teams are more sure of what descriptions correlate to which symbols and can factually state whichever symbol they have. This removes any subjective score from the messages they send. Again, there is no significant effect of the network on the overall subjectivity of each message. They all had similar trends of messages that neutralized as the game progressed. It is noted that the confidence intervals of the final trials are larger and is less accurate, but for the majority of the trials, the neutralization trend is still observed. With experience, team members share fewer subjective messages with emotional affect and send more concrete messages.

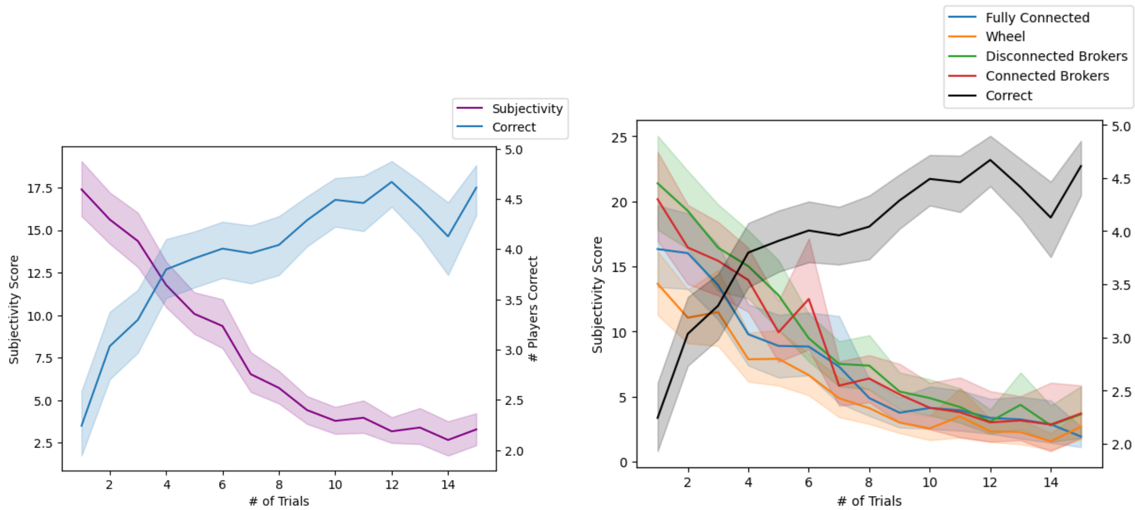

Figure 5-6: Cumulative subjectivity score vs Correct proportion. The cumulative subjectivity score within a trial decreases per trial as the proportion of players selecting the correct shared symbol increases per trial.

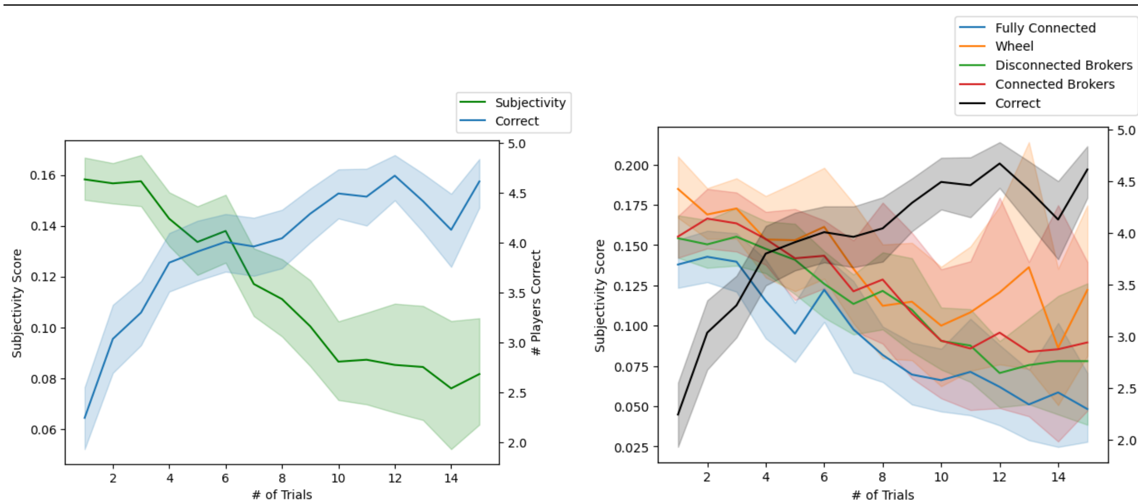

Figure 5-7: Subjectivity score per message vs Correct proportion. The average subjectivity score per message decreases per trial as the proportion of players selecting the correct shared symbol increases per trial.

5.3.3 Player and Game Reference Sentiments

As the game progressed, we also calculated the average polarity and subjectivity per message per trial using the messages that had player and game references. We not only wanted to observe the content of these messages, but also how the sentiment of these messages changed over time.

Player Reference Sentiments

Fig 5-8 shows that messages that referenced another player on average had neutral polarity. We expect that the majority of player referenced messages in the early trials are a mix of high polarity (i.e. greetings, excitement, and early success) and low polarity (i.e. confusion, frustrations, and early failures). This mix of both high and low polarity messages averages out to neutral as seen in Fig 5-9. As a team develops their TMS and refines their problem-solving framework, we expect that the player referenced messages are more focused on the current task and describing symbols, which are more neutral in polarity. Although the graph shows that the messages are mostly neutral, there is a difference in the content of the messages from the early trials and later trials. In Fig 5-9, it is observed that the proportion of neutral messages compared to positive and negative messages gradually increases as teams develop their TMS. In addition, the total positive and negative polarity scores decreases over subsequent trials as well. Overall, we again see that players are not only coordinated on the physical task, but they are also emotionally coordinated. As trials pass, players will tend to put aside any outbursts of emotions so that they can focus on the task and not disrupt the team processes.

Fig 5-10 shows that the subjectivity score of messages that referenced another player tended to slightly decrease over trials. As trials pass, players will send less opinionated messages, because the team would have developed a shared problem-solving framework that is mostly focused on communicating the symbol references, which has a low subjectivity, since they are normally simple nouns and verbs. When players have clarity of their own role and their teammates roles, they no longer need

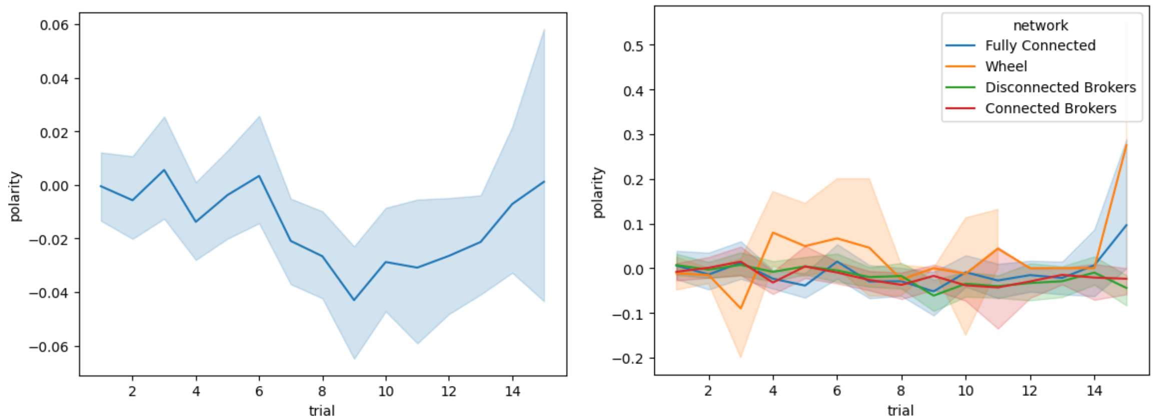

Figure 5-8: Player references polarity per trial. The messages that contained player references on average had a neutral polarity score throughout the trials.

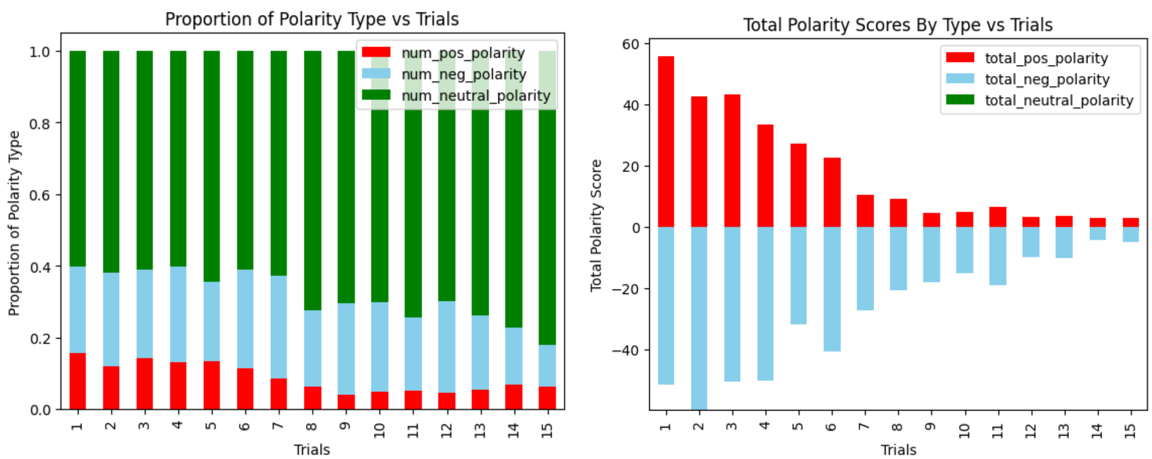

Figure 5-9: The figure on the left shows the proportion of positive, negative, and neutral polarity messages over all trials. The number of neutral polarity messages increased throughout the trials while the number of positive or negative polarity messages decreased. The figure on the right shows the total polarity scores over all trials. The cumulative positive and negative polarity scores decreased per trial. Within each trial, the cumulative positive and negative polarity scores combined to have a net score that was close to neutral.

to send subjective thoughts on what each player should be doing. The players are already coordinated within the team and have a clear picture of what is expected of each player.

Note that we did not have many messages with player references from the Wheel network, so the polarity and subjectivity trends for the Wheel network is less robust.

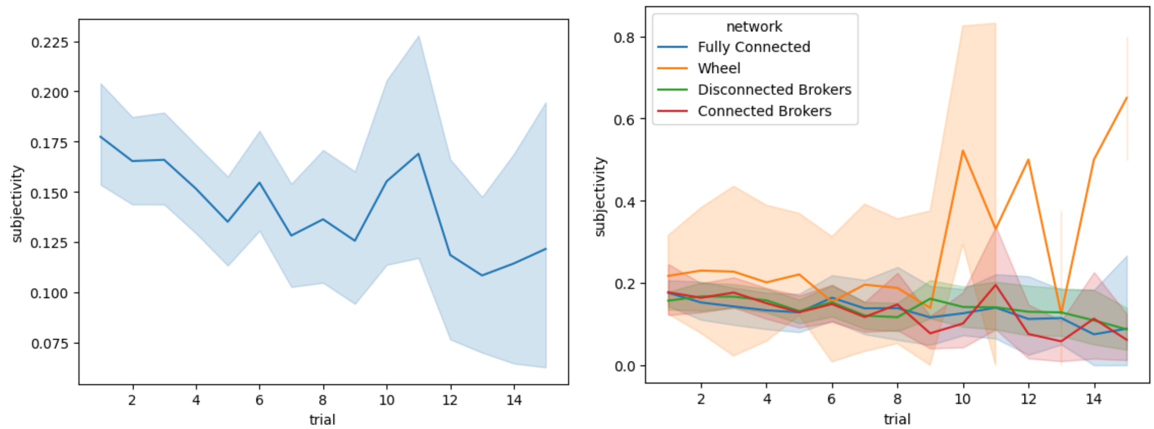

Figure 5-10: Player references subjectivity per trial. The messages that contained player references tended to decrease in subjectivity score per trial.

Game Reference Sentiments

Fig 5-11 shows that in general there is a negative polarity per message across the trials. We expect that players will reference the game if they want to voice their frustrations, confusions, or concerns about the game, especially when they are in the beginning stages of developing their problem-solving framework. These references will generally have a negative polarity score. If a player enjoys the game, they are unlikely to voice their thoughts throughout the game as voicing positive feedback has little contribution to a team's TMS when players are in the middle of playing to get as many trials correct as possible. However, players are likely to voice their thoughts about the game at the very end when the game comes to a close. If a team successfully developed a TMS, they are more likely to have found success in the trials and have more positive views about the game.

Fig 5-12 likewise shows that messages that include a game reference will have a positive subjectivity score. We expect that references to the games are likely to be a review or question about the game which are subjective as they include personal opinions or non-factual statements.

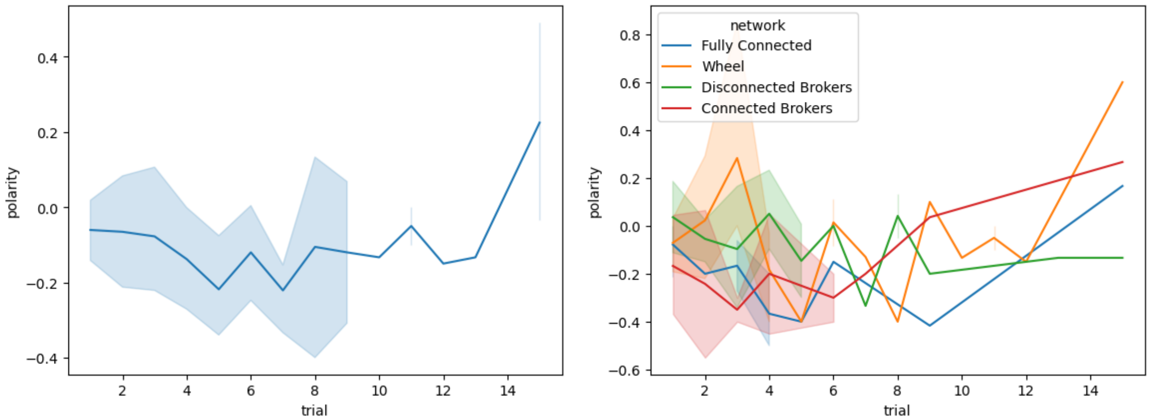

Figure 5-11: Game references polarity per trial. The messages that contained game references tended to stay at a constant negative polarity throughout all the trials. The only exception was near the final trials when the polarity would become positive.

Again, it is to be noted that there were not as many direct game or research references, so the trends are less robust than the player references.

5.4 Role Clarity and Coordination

A different way to measure role clarity and coordination is to examine the structure of the team through the distribution of messages among the team members. Within a team, we can discern specific roles by the amount of messages they send. Specifically, players that proportionally send more messages than everyone else are likely to be the de facto leader within the team. To quantify this distribution, we can analyze a team's Theil Index.

5.4.1 Theil Index

The Theil index T_T is the same as redundancy in information theory which is the maximum possible entropy of the data minus the observed entropy. It can also be viewed as a measure of lack of diversity, inequality, and non-randomness. The Theil index is derived from Shannon's measure of information entropy S, where entropy is a measure of randomness in a given set of information. The general form of observed

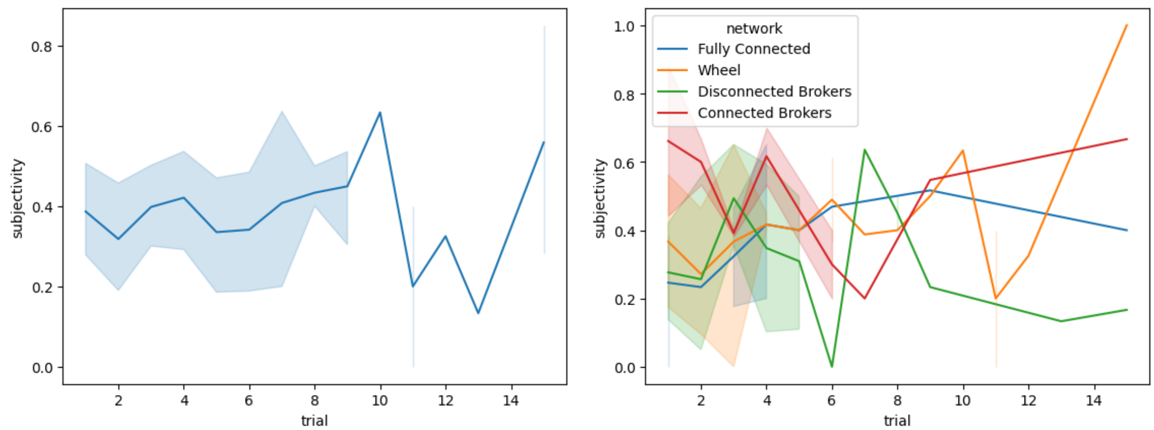

Figure 5-12: Game references subjectivity per trial. The messages that contained game references tended to stay at a constant positive subjectiity throughout all the trials.

entropy is

$$S_{Theil} = k\sum_{i=1}^{N}\left(p_i \ln\left(\frac{1}{p_i}\right)\right) = -k\sum_{i=1}^{N}(p_i \ln(p_i))$$

where

- i is an individual subject from the team
- p_i is the ratio of messages player i sends in a trial to the total number of messages sent in a trial
- k is a constant, typically equal to 1 in information theory or statistics

The Theil index T_T measures how far the observed entropy (S_{Theil}, which represents how randomly messages are distributed) is from the highest possible entropy ($S_{max} = \ln(N)$, where N is the number of players in the team). S_{max} represents messages being maximally distributed amongst all individuals, which equates to an equal distribution of all messages.

$$T_T = S_{max} - S_{Theil} = ln(N) - S_{Theil}$$

As subsequent trials pass, Fig 5-13 shows that teams gradually have a higher Theil

Index which indicates a higher inequality in the distribution of messages. A higher inequality in the distribution of messages can correlate to the emergence of leaders within a team, as team leaders tend to dictate their subordinates more by sending messages on the information they have as well as dictating what the subordinates should do. Looking at the average Theil indicies of the different network structures, it's clear that the Wheel network has the highest Theil index or the most unequal distribution of messages amongst players. From the very beginning, the Wheel network has a central player that must coordinate all other players within the team which means receiving and sending the most number of messages. Following the Wheel network are the two variations on the Wheel network, the CB and DB network. Similar to the Wheel, the CB and DB has two defined leaders or brokers that are connected to more people than the other players. In the case of the CB network, the brokers are directly connected to each other which allows them to share their relational information and have a clearer picture that they are the central leaders of the two independent 3-player cliques. Therefore, the CB is seen to have a higher Theil Index than the DB network. On average, the Clique network has the lowest Theil Index, since it's designed to allow all team members to talk to each other which in theory better promotes an equal distribution of messages. However, it's observed that near the final trials of the game, we can see a glimpse emergence of leaders to help better coordinate all of the information that's flowing through the different channels. As subsequent trials pass, teams are able to coordinate their roles in the team and understand if there are any main leaders within the team. If there are leaders, teams can coordinate around the team leader to combine all of the information. If there are no leaders, players are able to share their successes and failures and discover productive combinations of their individual knowledge and expertise

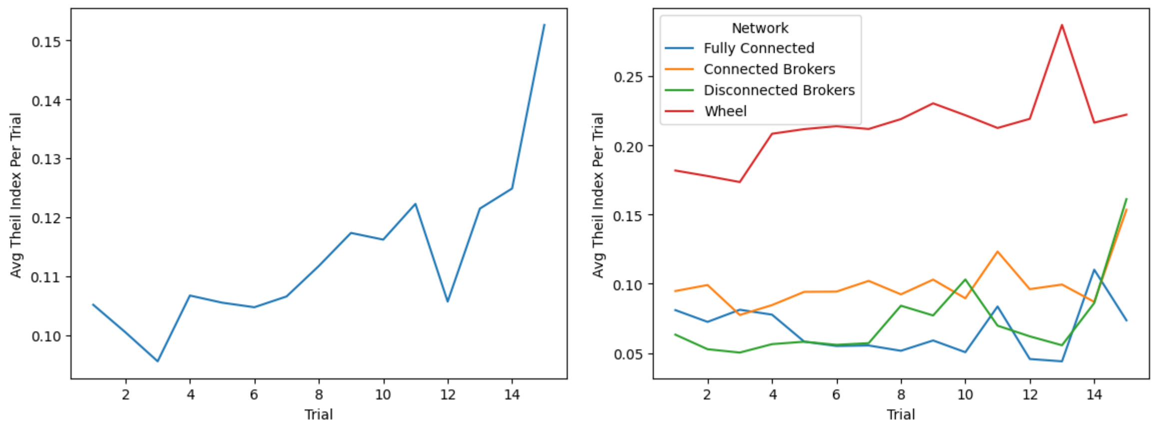

Figure 5-13: Average Theil Indicies per trial. The average Theil Index within a team gradually increases which indicates a higher inequality in the distribution of messages and the emergence of team leaders.

Chapter 6

Deeper Dive into NLP Methods to Extract Team Processes

6.1 Topic Modelling

We want to test the potential of a method called Discourse Atom Topic Modeling (DATM) to identify topics in our communication data corpus and represent messages as topic sequences. DATM draws on advances in theoretical machine learning to integrate topic modeling and word embedding, capitalizing on their distinct capabilities. Discourse atoms can be interpreted as latent topics; through a generative model, atoms map onto distributions over words. We can also infer the topic that generated a sequence of words (Foster et al., 2022). [2]

6.1.1 Integrating Topic Modeling and Word Embedding

To examine if any topics or idea clustered emerged from the communication data, I integrated topic modeling and word embedding together. To integrate topic modeling and word embedding, we address two core challenges. First, we identify latent topics in a trained word embedding space. In our experiment, we set out to identify topics in an embedding space trained on the communication data from the symbol task games. Second, we identify the topics underlying an observed set of words (e.g. a message or a shared language). More generally, we need a theoretical framework to connect an embedding space to raw text data. DATM integrates several methodological and theoretical advances in research on word embeddings to address these two challenges, as described next.

First, I created a word-embedding trained on the team communication data accumulated across all games. To identify topics in this embedding space, I apply K-singular value decomposition (K-SVD) to the word vectors. K-SVD is a sparse dictionary learning algorithm that outputs a set of K vectors (also dubbed "discourse atoms") such that any of the V word vectors in the vocabulary can be written as a sparse linear combination of atom vectors. The words refer to broader topics and a message is a combination of words from topics. To measure the quality of topics, I use a combination of proposed metrics for topic model quality and R^2, an additional metric for K-SVD. These metrics quantify how internally coherent the topics are, how distinct they are, and how well the discourse atoms explain the word-embedding space itself. We apply K-SVD to our word embedding while varying the number of atom vectors K. To select a final sparse representation, we use a combination of previously proposed metrics for topic model quality. The number of discourse atoms you decide to make is largely determined by the balance between the product of R^2 and topic diversity. R^2 tends to increase with a higher number of topics or as topic diversity decreases. The intuition behind balancing the number of discourse atoms is that more topics can better explain the original semantic space. However, these topics are then less distinct from one another. Thus, we want a balance between these two. We select our final model with 8 atoms to balance performance across these metrics. This results in 8 atoms each with 25 topics totaling to 200 topics.

Sparse dictionary learning offers a way to identify the topics of semantic space, but it does not map observed sequences of words (messages) to these topics. Fortunately, a recently proposed language model offers a link between messages and points in our topic space: the latent variable model. This model provides a simplified, probabilistic account for how the text in our communication data corpus was generated. But it also provides a theoretically motivated algorithm to summarize a given message as a message vector in the word-embedding space (a message embedding). For a given message vector, we can find the closest atom vector in word-embedding space and thus map observed sequences of messages to latent topics. For each game, we assign each message vector in a sequence of message vectors (the entire game) to a topic. This yields a sequence of latent topics that represents the game. (Arseniev-Koehler et al., 2022) [3]

Looking closer at the latent variable model, the probability that a word w being present at some location t in the corpus is based on the similarity of its word vector $\mathbf{w}$ and the latent "gist" at that point in the corpus $\mathbf{c}_t$ (discourse vector). The word most likely to appear at t is the word most similar to the current "gist". The similarities between possible word vectors, $\mathbf{w}$, and the discourse vector, $\mathbf{c}_t$, can be turned into a probability distribution over words by first exponentiating the similarities ($exp(\langle \mathbf{c}_t, \mathbf{w} \rangle)$). Then, we divide by their sum Z_{c_t}, so that the distribution sums to 1. This model thus associates a distribution over words to every point in word-embedding space. It also sets up a relationship between atom vectors (points in word-embedding space) and topics. The model treats corpus generation as a dynamic process, where the t-th word is produced at step t. The process is driven by the random walk of a gist through the word-embedding space. At each step t, a word is generated. This process is defined by Eq 6.1.

$$\Pr[\mathbf{w} \text{ emitted at } t|\mathbf{c}_t] = \frac{exp(\langle \mathbf{c}_t, \mathbf{w} \rangle)}{Z_{c_t}} \tag{6.1}$$

To make the model more realistic, we include the overall frequency of the word in the corpus, $p(w)$, the local gist (c_t) and the global gist (c_0). The global gist represents the overall syntactic and semantic structure of the corpus. The relative importance of the word frequency and the gist is controlled by α and the relative importance of the local and global gist is controlled by β (Arora et al., 2016) [1]. This improved latent variable model is defined in Eq. 6.2.

$$\Pr[\mathbf{w} \text{ emitted at } t|\mathbf{c}_t] = \alpha p(w) + (1-\alpha)\frac{exp(\langle \tilde{\mathbf{c}}_t, \mathbf{w} \rangle)}{Z_{c_t}}, \tag{6.2}$$

where $\tilde{\mathbf{c}}_t = \beta\mathbf{c}_0 + (1-\beta)\mathbf{c}_t$ and $\mathbf{c}_0 \perp \mathbf{c}_t$

In the generative direction, Eq. 2 fixes the probability of a word appearing, given details of the communication corpus and the current gist. In our experiment, however, we observe the words and the gist is latent. In DATM, we want to infer the gist (i.e., where we are in semantic space) given an observed set of context words and then map this gist to an atom vector. Given the generative model and the smooth inverse frequency (SIF) weighting scheme, we can compute the maximum a posteriori (MAP) estimate of the combined context vector for a set of context words $\mathcal{C}$ using Eq. 6.3.

$$(\tilde{\mathbf{c}}_t)_{\text{MAP}} = \sum_{w \in \mathcal{C}} \frac{a}{p(w)+a}, \text{where } a = \frac{1-\alpha}{\alpha Z} \tag{6.3}$$

$(\tilde{\mathbf{c}}_t)_{\text{MAP}}$ is a weighted average of the word vectors in the context window. Using SIF, words are weighted based on their corpus frequency $p(w)$. Frequency words make a smaller contribution to the estimate of c_t. For a given set of context words $\mathcal{C}$, we can estimate c_t by using Eq 3. to compute $(\tilde{\mathbf{c}}_t)_{\text{MAP}}$ from word vectors in $\mathcal{C}$ and then subtract off its projection onto c_0. $(\tilde{\mathbf{c}}_t)_{\text{MAP}}$ then becomes an estimate of the latent local gist of $\mathcal{C}$, as a point in word-embedding space.

We have combined three methodologies — sparse coding of the embedding space, the latent variable model, and SIF sentence embeddings — into a cohesive procedure that allows us to discover latent topics in our communication corpus and to identify the topic that best matches the estimated gist of an observed message. Finally, to infer topics across all of the game, we estimate the gists all of the messages in corpus. This last step yields the sequence of topics underlying the communication data. Taken individually, each component of DATM offers an effective tool for extracting the shared content from the communication data.

6.1.2 Word Clustering

In order to compare the relevancy of the words within a discourse atom, we used a word clustering technique to cross-check the discourse atom results. First, we took all the words outputted by the discourse atoms, converted them into word vectors

```
Out[421]: <AxesSubplot:title={'center':'Discourse Word Clusters'}, xlabel='tsne-2d-one', ylabel='tsne-2d-two'>
```

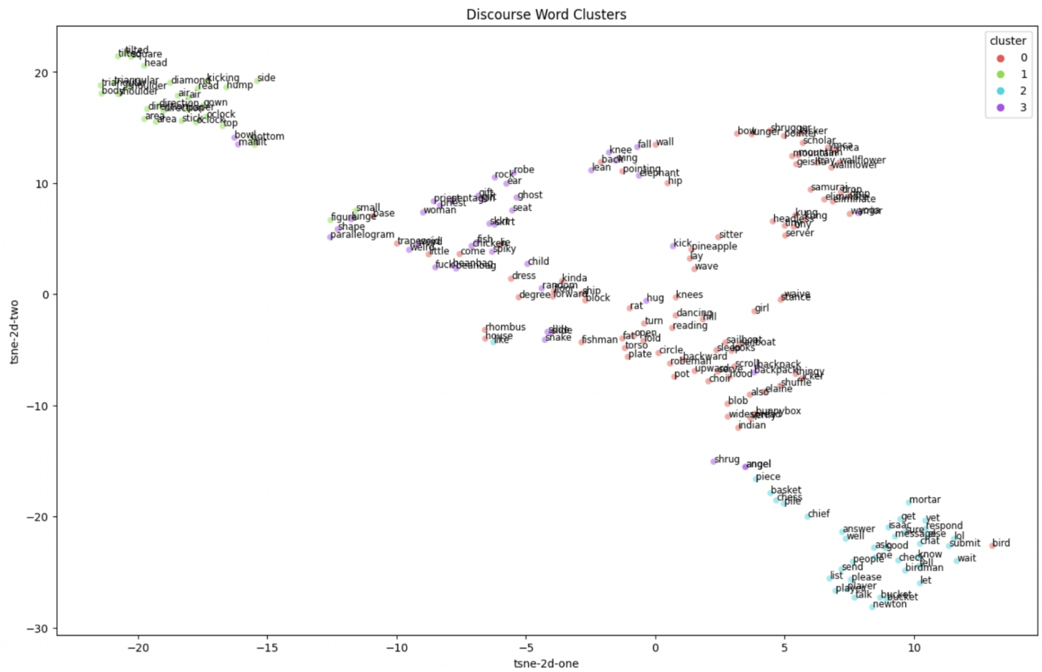

Figure 6-1: Word Clustering with 4 word clusters. Cluster 0 and 3 were mixed together within the middle of the graph. Cluster 1 was clustered in the top-left of the graph. Cluster 2 was clustered in the bottom-right of the graph.

using a word embedding dictionary, and placed them all into a single matrix. This word vector matrix was then trained on multiple Gaussian Mixture Models (GMM) with varying component sizes to find the number of clusters that best fit the matrix.

In order to visualize the word clusters, we first passed the discourse word vector matrix into a Principal Component Analysis (PCA) to reduce the dimensions of the matrix. Then, we took the top 2 most principle components of the PCA output and passed that through a t-Distributed Stochastic Neighbor Embedding (t-SNE). The t-SNE output can then be graphed and colored to show the word clusters. One such word clustering graph can be seen in Fig 6-1.

6.1.3 Comparing Discourse Atoms and Word Clusters

Fig. 6-2 shows 8 discourse atoms that were produced from training on the team communication data. Each word in the discourse atoms are colored according to the word clustering assigned by the GMM. Some significant atoms that we found were Discourse Atoms 5 and 6.

Discourse Atom 5 was predominantly made up of words from the green-colored clustering which is seen in the top-left of Fig 6-1. These words are related in that they represent a strategy revolved around a symbol description strategy in which description such as those focused around different shapes (square, diamond, triangular) and the locations of each shape (top, bottom). This tactic often correlated with longer messages in which players directly described different shapes within the tangrams, the number of edges, and the location of each shape in relation to other shapes. This Discourse Atom successfully extracted a prime strategy used within teams.

Discourse Atom 6 was made up of words from the blue-colored clustering which is seen in the bottom-right of Fig 6-1. These words are related in that they are communication words which are used to clarify misunderstandings and provide guidance to other players. This is yet another team process. This Discourse Atom provides a list of commonly used words that players used to coordinate with each other on retrieving information from other players, updating other players on the current states of other players, and submitting their answers.

The rest of the Discourse Atoms are predominantly made up of words from the red and purple-colored clustering which is seen in the middle of Fig 6-1. The red and purple clusters are less distinct than the blue and green clusters and can be seen to be dispersed among the other Discourse Atoms. However, these Discourse Atoms do represent another strategy revolved around assigning symbols a specific labels or phrases to refer to them by. These labels are often more abstract and require some inference to figure out. This strategy is another kind of team process, similar to the symbol descriptions. We suspect that the diversity of symbol references causes the topic model to have difficulty in grouping the labels. This results in the few discourse atoms to be combinations of words from different clusters, but the general topic of each atom is evident. One team may use a word to describe a certain symbol, whereas another team may use a similar or the same word to describe a completely different symbol. The ambiguity of the tangram symbols is a virtue of the diverse set of solutions within the problem solving space of this game. Teams can find an infinite number of solutions in terms of strategy as well as ways to identify symbols, yet this discourse atom model can identify and extract important topics within the team data.

6.1.4 Linking Discourse Atoms to TMS and Team Performance

Using the idea that Discourse Atoms provide keywords from different aspects of team processes, we can then correlate those team processes to how they affect a team's TMS and team performance. Fig 6-3 shows the proportion of words from each cluster as subsequent trials pass. In the initial trials, there were more words from cluster 1 and 2, which were predominant in Discourse Atom 5 and Discourse Atom 6. We can see that in the early trials, before a team's TMS develops, the team uses more communication and coordination words to make sure that everyone is actively talking with each other, figuring out who they should message and respond to, and keeping everyone on the same page. Similarly, before a team's TMS develops, teams are more likely to use symbol descriptions to describe symbols to each other and to clarify which symbols players are talking about. As trial pass and players develop their shared problem-solving framework, we observe that the use of these cluster words decreases. Instead, words from clusters 0 and 3 become more prominent within the trial. These cluster words are specific label or phrases that teams use to reference specific symbols. As a team develops a shared language, they can more efficiently and effectively describe the symbols that they have and coordinate which symbol they have in common. In terms of team performance, we saw that teams that developed a shared problem-solving framework and a shared language often performed better than those that didn't.

6.1.5 Future Work on Discourse Atoms

We believe that there is a lot of potential in using Discourse Atoms in discovering different team processes. The nature of this unstructured task allow for too diverse of

Discourse Atom 0
wave, tray, bird, oclock, kung, skirt, paper, read, lay, direction, bunnybox, waive, air, kick, bucket, gown, backpack, scroll, lie, gift, server, open, fall, warrior, sailboat,

Discourse Atom 1
spiky, shoulder, backpack, priest, weird, chess, piece, skirt, triangular, ghost, direction, beanbag, slide, chief, child, sailboat, gift, oclock, tilted, air, random, rock, shape, area, indian,

Discourse Atom 2
pentagon, shrug, fish, yoga, fishman, stance, basket, pile, bucket, newton, torso, also, robeman, birdman, bowl, verify, robe, isaac, angel, man, parallelogram, player, mortar, lean, knee,

Discourse Atom 3
elephant, geisha, chicken, scholar, pineapple, mountain, beanbag, fuck, woman, hug, ymca, seat, wallflower, drop, priest, tiny, eliminate, slide, ear, lunge, samurai, snake, angel, weird, kung,

Discourse Atom 4
bow, shrugger, elaine, pointer, lunger, icker, kicker, wall, hip, thingy, back, sitter, pointing, shuffle, headless, wing, mountain, ymca, wallflower, side, drop, tiny, girl, eliminate, knees,

Discourse Atom 5
top, body, square, area, diamond, triangular, base, trapezoid, stick, tilted, bottom, little, small, head, shoulder, widespread, come, house, like, tilt, figure, direction, rhombus, hump, kicking,

Discourse Atom 6
talk, please, list, tell, send, people, let, check, know, well, one, ask, wait, player, good, sure, else, lol, chat, message, get, submit, answer, respond, yet,

Discourse Atom 7
fold, forward, fat, rat, backward, dancing, looks, floor, kinda, choir, degree, block, reading, plate, serve, sleep, ship, upward, pot, circle, hood, turn, blob, hill, dress,

Figure 6-2: 8 Discourse Atoms Results. Using 8 discourse atoms, we saw distinctions within discourse atom 6 and discourse atom 5. Discourse atom 5 modelled words from cluster 1 and discourse atom 6 grouped modelled from cluster 2. The other atoms were mixed with words from cluster 0 and 3.

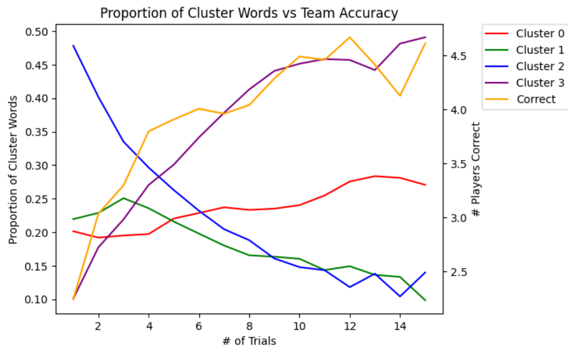

Figure 6-3: Proportion of cluster words per trial vs proportion of correct symbol selection. As subsequent trials passed, the proportion of words from clusters 1 and 2 decreased while the proportion of words from clusters 0 and 3 increased. This coincided with the proportion of players that selected the correct symbol increasing.

a solution space which makes it difficult for the topic model to find distinct models. Despite the sparse, yet diverse corpus that this topic model trained on, it was still able to distinguish three important aspects of team processes: symbol descriptions (shapes and shape locations), symbol references (labels), and coordination messages. If the task was more structured or more constrained in the number of solutions, this topic model method may yield more structured topics and results.

6.1.6 Future Work on Problem Solving Space Analysis

With many different teams solving the same unstructured task, there will be a diverse set of solutions that teams develop as they learn from their respective successes and failures. It would be interesting to map each team's solution in a larger problem solving space and analyze if any of the teams converge to the same solution as they go through the tasks. If they do, we could analyze what aspects of their team processes contributed to their solution or team performance.

Appendix A

Tables

Appendix B

Figures

Bibliography

[1] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016.

[2] Alina Arseniev-Koehler, Susan D. Cochran, Vickie M. Mays, Kai-Wei Chang, and Jacob G. Foster. Integrating topic modeling and word embedding to characterize violent deaths. *Proceedings of the National Academy of Sciences*, 119(10):e2108801119, 2022.

[3] Alina Arseniev-Koehler, Susan D. Cochran, Vickie M. Mays, Kai-Wei Chang, and Jacob G. Foster. Integrating topic modeling and word embedding to characterize violent deaths. *Proceedings of the National Academy of Sciences*, 119(10):e2108801119, 2022.

[4] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155, 2003.

[5] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information, 2016.

[6] Ronald S. Burt and Ray E. Reagans. Team talk: Learning, jargon, and structure versus the pulse of the network. *Social Networks*, 70:375–392, 2022.

[7] Herbert H. Clark and Deanna Wilkes-Gibbs. Referring as a collaborative process. *Cognition*, 22:1–39, 1986.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[9] Peter Foltz and Melanie Martin. Automated communication analysis of teams. *Team Effectiveness in Complex Organizations, Taylor Francis Group, New York, NY*, pages 411–431, 2009.

[10] Peter Foltz, Melanie Martin, Ahmed Abdelali, Mark Rosenstein, and Rob Oberbreckling. Automated team discourse modeling: Test of performance and generalization. *In Proceedings of the Annual Meeting of the Cognitive Science Society*, 28:28–33, 11 2006.

[11] Jamie Gorman, Melanie Martin, Terri Dunbar, Ronald Stevens, and Trysha Galloway. Analysis of semantic content and its relation to team neurophysiology during submarine crew training. 8027:143–152, 07 2013.

[12] H.J. Leavitt. Some effects of certain communication patterns on group performance. *The Journal of Abnormal and Social Psychology*, 46(1):38–50, 1951.

[13] Kyle Lewis. Measuring transactive memory systems in the field: scale development and validation. *Journal of applied psychology*, 88(4), 2003.

[14] Michelle Marks, John Mathieu, and Stephen Zaccaro. A temporally based framework and taxonomy of team processes. *The Academy of Management Review*, 26:356, 07 2001.

[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[16] Alex "Sandy" Pentland. The new science of building great teams. *Harvard Business Review*, 2012.

[17] Ella Miron-Spektor Ray E. Reagans and Linda Argote. Knowledge utilization, coordination, and team performance. *Organization Science*, 27(5):1108–1124, 2016.

[18] Yuqing Ren and Linda Argote. Transactive memory systems 1985–2010: An integrative framework of key dimensions, antecedents, and consequences. *Management Annals*, 5(1).

[19] Ranjani Krishnan Richard L. Moreland, Linda Argote. Socially shared cognition at work: Transactive memory and group performance. 1996.

[20] Hagay C. Volvovsky Ronald S. Burt, Ray E. Reagans. Network brokerage and the perception of leadership. *Social Networks*, 65:33–50, 2021.

[21] Randall D. Spain, Michael Geden, Wookhee Min, Bradford W. Mott, and James C. Lester. Toward computational models of team effectiveness with natural language processing. 2019.

[22] Xuan-Son Vu, Thanh Vu, Son Tran, and Lili Jiang. Etnlp: A toolkit for extraction, evaluation and visualization of pre-trained word embeddings, 03 2019.

[23] Bei Yan, Andrea B. Hollingshead, Kristen S. Alexander, Ignacio Cruz, and Sonia Jawaid Shaikh. Communication in transactive memory systems: A review and multidimensional network perspective. *Small Group Research*, 52(1):3–32, 2021.