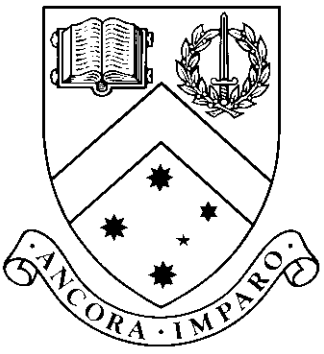
Extending a Model for Simulating Collaborative Discourse

by

Robert Brian Milligan (30579643)



Final Thesis - 5373 words (FIT5126-8 18 Credit Points)

Submitted by Robert Brian Milligan (30579643)

for partial fulfillment of the Requirements for the Degree of

Master of Data Science (C6004)

Supervisor: Lecturer Zachari Swiecki

Faculty of Information Technology
Monash University

May, 2024

Extending a Model for Simulating Collaborative Discourse

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Robert Brian Milligan (30579643)
May 22, 2024

Table of Contents

Extending a Model for Simulating Collaborative Discourse

Student ID: 30579643

Name: Robert Brian Milligan

Course: FIT5126 (18 Credit Points)

Supervisor: Zachari Swiecki

Contents

1. Introduction

The study of collaborative learning as an academic discipline has been explored since the early 20th century from Lev Vygotsky's contribution of the zone of proximal development in the 1930s (Yasnitsky, 2018), which suggests that with the aid of teacher or expert, the range of tasks a learner can complete is greater. Since then, this discipline has been built upon by multitudes of researchers who try to understand how groups or people work together, learn, share information, and attempt to solve problems as a team. More recently, researchers have begun to apply computational methods to study collaboration (Boulay et al., 2023). The study of collaborative learning is of great benefit to many areas especially in education but also with applications in the military (Swiecki et al., 2020), research and development (Swiecki et al., 2022) and healthcare (Wooldridge, 2018), among many others. Understanding how collaborative processes occur is beneficial as it can help leaders and managers foster teams to improve task performance and learning,

When studying collaborative learning, researchers typically collect data in the form of discourse. Discourse is a set of behaviours (such as utterances) that can be analysed and can be from a large variety of different scenarios. Notes of discussions, transcripts of interviews, email exchanges and audio recordings are all forms of discourse. Performing these experiments can be expensive and time consuming, requiring the gathering of data between groups of participants for an extended period of time. More recently the computational generation of discourse data has been explored as a promising approach. This could assist researchers designing collaborative learning experiments. The benefits of this would not just be financial but may also be used to design experiments researchers would not have thought of conducting otherwise, to test hypotheses they have developed when experimenting with simulated data.

This literature review has four aims. First, I look at explaining various themes relevant to the analysis of collaborative learning. Second I compare the various methods of analysing collaborative data. Third, I investigate the various methods of simulating collaborative data and leads to an exploration of the variety of coded datasets that are used in collaborative discourse. Ideas common to multiple papers are explained and themes from the literature synthesised. Finally, I discuss gaps in the current literature. One of these gaps will form the basis for a research project. Namely, that within the existing models for simulating discourse datasets they lack the ability to model behaviours that evidence more than one topic. Creating a solution to this limitation would help the simulated data to match the dynamics found in real world data more closely and expand the types of collaborative discourse this model could be used to simulate.

The document will end with a research project plan that will outline the aims of the research project and an exploration of the scope the solution will need to achieve. Moreover it will outline a few ideas for how a solution could be developed with a preliminary procedure and a timeline for when key stages of the project are completed and when milestones are due. The project is shown to be in the given timespan as an existing model to extend has been identified and multiple methods of addressing the limitation of the existing model have been designed.

2. Substantive Literature Review

Coded Data in Collaborative Discourse

One of the main basis for computationally analysing discourse data is a coded data dataset as seen in figure 1. In this case, the discourse is in the form of a chronological list of turns of talk or *utterances*. Almost always an utterance is verbal but in some cases it can rely on other inputs such as logs (Paquette et al., 2021) or sensor readings (Andrist et al., 2015). Usually, but not always, an utterance is received by everyone in the group even if the speech in the utterance itself is directed only at a certain individual. In the study of collaborative discourse this is known as a wide "horizon of observation" (Hutchins, 1996).The utterances in the data can be split up numerous different ways from single sentences to an entire speech. Each utterance will have associated with it a list of *codes* that were applied to the data. Codes can represent a wide range of concepts but ultimately classify the utterances to gain further insights from them (Shaffer & Ruis., 2021). The main goal of coding is to extract the meaning or to categorise an utterance in some way. For example, in coded data shown in figure 1, each of the columns Technical through to SE is a Code. In this dataset and many others null means that the corresponding utterance did not relate to any of the codes which are being considered.

UserName	text	line.id	Technical	Performance	Client.and.Consultant	Electric_c	Hydraulic_c	PAM_c	Pneumatic_c	SE_c	null_c
akash v	I think the electric actua	78	1	0	0	1	0	0	0	0	0
akash v	I will have the medical s	79	0	0	1	0	0	0	0	0	0
alexander b	I agree with Brandon. ne	80	0	0	0	0	0	0	0	0	1
alexander b	I will take Benjamin Tayl	81	0	0	1	0	0	0	0	0	0
brandon l	Meredith Yamasaki-Nola	82	1	0	0	0	0	0	0	0	0
brandon l	Meredith Yamasaki-Nola	83	1	0	1	0	0	0	0	0	0
akash v	@Brandon: Oh wait I see	84	0	0	0	0	0	0	0	0	1
akash v	I'm kind of confused on	85	0	0	1	0	0	0	0	0	0
brandon l	@Akash: The email seer	86	1	0	0	0	0	0	0	0	0
brandon l	@Akash: The email seer	87	1	0	0	0	0	0	0	0	0
akash v	So we pretty much do th	88	1	0	0	1	0	0	0	0	0
akash v	@Justin Kim: ok thanks	89	0	0	0	0	0	0	0	0	1
akash v	Does it talk about the RF	90	1	1	0	0	0	0	0	0	0
brandon l	@Akash: I didn't see any	91	0	1	0	0	0	0	0	0	0
brandon l	@Akash: I didn't see any	92	0	1	0	0	0	0	0	0	0
brandon l	So we need to do the pr	93	0	0	0	0	0	0	0	0	1

Fig 1. A set of coded data used in part of Swiecki et al. (2022)

Analysis of Collaborative Discourse Data

Coded datasets allow researchers to investigate collaborative problem-solving, which is theorised to contain both social and cognitive components. For example conflict resolution through negotiation (Griffin et al., 2018). Similarly knowledge building is the concept where multiple people contribute their own skills or knowledge which are combined to work on a shared problem (Griffin et al., 2018).

One way that researchers analyse the socio-cognitive nature of collaboration is using process models. For example, Katz et al. (2000) used machine learning to discover the discourse structure to look at interactions between students and teachers. Another method is to analyse the connections between codes is to segment that data either by counting connections between codes in the entire conversation or by iterating through utterances and marking those that appear close to each other time, this is known as a moving stanza or moving window (Siebert-Evenstone et al., 2017). This process of segmenting data can be continued by performing Epistemic Network Analysis (ENA), a technique that takes segmented data, identifies the co-occurrence of codes within those segments and represents aggregated co-occurrence networks in a low-dimensional embedding space. For example, Swiecki (2021) used ENA to examine the collaborative talk of military officers in training.

Collaborative data can take many forms that which may affect analysis. Two main forms are if the coded data is analysing the content or themes in an utterance. In comparison to coded data which also encodes the structure about how the utterance relates to others to to who they are directed to.

Examples of the former tend to classify the sentiment or the purpose of an utterance. Jackowska (2021) looked at a virtual team of IT workers working together on projects as a team. The codes focused on the purpose of various utterances for example “question”, “agreement”,”disagreement” and “elaborating”. Similarly, Shaffer et al. (2016) used text data from RescuShell a virtual internship of small groups working on a R&D project. The codes focused again on the context of messages for example “materials”,”sensor” and ”design”. Oner (2020), Swiecki et al. (2020), Shah et al. (2021) and Paquette et al. (2021) all used codes in a similar way looking at the themes or the types of speech used by agents that broadcasted their utterances to a group and applied ENA in their analysis.

In comparison, Brohinsky et al., (2021) while using the same type of coded dataset applied a different form of ENA for their analysis, they discussed what they called a trajectory model. Similarly to standard ENA this places codes on one axis however the second axis contains temporal information, in their example of Romeo and Juliet, themes like fear and love were on the x axis while the y axis used a temporal element, showing the act of the play. Usually to display a temporal element multiple plots are shown on the same axis for example shown in Larson et al (2021).

Kapur et al. (2010) used a different approach to analyse their discourse data. They were looking at sets of conversations between pairs where there were disagreements involving a misconception. Each utterance was marked as having a positive or negative impact towards coming to the truth and could be represented as a markov random walk. In comparison to other models that used codes, this model assigned a number of “-1” or “1” modifiers to each utterance to represent a movement on a markov random walk. This is still a form of coding the data as each line would have a certain number of “-1” and “1” associated with it. Using data from 20 different conversations they were able to predict the eventual outcome of future conversations after observing the first 30-40% of the discussion.

In contrast, some studies used codes to perform a structural analysis of the connections between various participants. Wooldridge et al. (2018) looked at a healthcare setting with various people such as doctors, nurses and clerks who would assign each other tasks. Four distinct parts of each utterance were tracked, the sender, receiver, if the task given was synchronous or asynchronous and as well as if the task was accepted or rejected. The study used codes such as “physician sender”, “nurse receiver”,“unit clerk sender”, “synchronous” and “asynchronous”. This coded dataset uses multiple codes to explain the purpose and direction of a message, rather than the underlying content itself. While also looking at the structure, Andrist et al. (2015) used codes in a method not usually seen, in comparison to speech or text, the coded data was collected using sensors which tracked the gaze of participants to monitor what they were looking at, at any time. An instructor explained to a worker how to construct a sandwich, looking and talking about various ingredients laid out between them. Examples of some of the codes tracked include “instructor gazing at the worker”, “instructor gazing at reference ingredient”, “worker gazing at non reference ingredient”, “worker gazing at a different object than instructor”. In both these papers the codes relate to each other and form a structure which is significantly different to content analysis.

Analysis of collaborative data	Content analysis	Structural analysis
Explicitly uses codes	(Oner, 2020) (Shaffer et al., 2016) (Swiecki et al., 2020) (Jackowska 2021) (Shah et al., 2021) (Paquette et al., 2021) (Brohinsky et al., 2021)	(Wooldridge et al.,2018) (Andrist et al., 2015)
Implicitly uses codes	(Kapur et al., 2010)	

Fig 2. A table summarising the various types of analysis that is performed on coded collaborative data

Simulation of Collaborative Discourse Data

While collecting, coding, and analysing data from collaborative settings has yielded insights into collaborative learning, this process can be difficult to implement due to the high costs of recruiting a large number of participants and collecting data. Because of this, some researchers have proposed simulating data such as Swiecki et al. (2022). One way to simulate discourse data is via *agent-based models*. Agent based models are a useful approach to study social phenomena as they incorporate ideas that can mirror the randomness of human nature. Agent based models are stochastic, making use of a random number generator to make various decisions. Moreover agents are usually heterogeneous and will use aspects about them to interact with the world differently, this can be thought of as each agent having their own behaviour (De Marchi., 2014)

For example, Park et al. (2023) used natural language processing and agent based modelling to create a simulated society of agents that could interact with another, have a memory of past conversations and engage in conversations directly with other agents using sentences. Social networks were created as a result in addition to the actual text that was transmitted between agents. This model was highly specific to the created world and agents. A paragraph of information about each agent was written and using natural language processing it was transformed into memories for each agent which would impact how they would interact with the world.

In addition, both Koponen & Nousiainen (2018) and Hamill & Gilbert (2010) created models that did not simulate text or codes but had agent based models that generated various structural connections between agents. The work of Koponen & Nousiainen looked at individual conversations between agents that were directional while the simulation Hamill & Gilbert created entire unidirectional social networks of agents that communicated with each other. Hutchins (1996) looked at a simulation where multiple agents talk to each other about 6 various different interpretations of an event, over time each agent can change their stance on each interpretation by discussing it with other agents.

Finally, Swiecki et al. (2022) created a model which could simulate the codes that would appear from analysing a real dataset. Its process was multistep and contained two distinct processes. Firstly a dataset is analysed and used to create two transition matrix types. Firstly a single matrix to simulate the order of discussion in a set of coded data and secondly for each speaker a matrix to simulate a single code which they broadcast to the rest of the group based on what was said previously in the discussion. This process does not attempt to simulate sentences but instead the patterns and dynamics of codes said by agents as seen in the input data.

Of the various models those which were most sensitive to the input data were of Park et al,. (2023) and Swiecki et al., (2022) which allowed for inputs of paragraphs of text to explain an agent's memories and a set of coded data respectively. The other three datasets created simulations for one single specific dataset to either simulate social networks, directions of interactions or the effects of interactions. A large section of collaborative data uses utterances which have been categorised with codes, therefore the model of Swiecki et al. (2022) would be the most useful for simulating that type of data.

Simulating Data with Agent based models	Simulating Content	Simulating Structure
Emphasis on Text Content	(Park et al., 2023)	(Park et al., 2023)
Emphasis on Code Content	(Swiecki et al., 2022)	
Neither Text or Code Content	Not Found	(Koponen & Nousiainen, 2018) (Hamill & Gilbert, 2010) (Hutchins, 1996)

Fig 3. A table summarising the various papers that categorises the simulation of collaborative data with agents based models.

Distribution of Codes in Coded Collaborative Discourse Data

As seen in previous sections, coded data makes up a large set of the types of data found in collaborative discourse data. Figure 4 investigates the average distribution of codes found in datasets as well as the extremes. Figure 5 tabulates the number of codes found in a dataset as well as the number of different codes each behaviour or utterance needs in order to simulate 95% and 99% of possible behaviours contained in the datasets. Finding open access data for collaborative discourse datasets is challenging due to the privacy of the data collected from participants. Two data sources were found, Swiecki et al. (2022) which contained conversational data of 12 groups and data provided by the International Society for Quantitative Ethnography. (n.d.) contained 5 different datasets of varying sizes. All datasets found were those that explicitly used codes and would be suitable for content analysis. Structural analysis would be possible to perform on them but would require additional codes to be added to the raw utterance data and be filled in by analysing the raw utterance text.

Figure 4 shows that depending on the dataset one is trying to simulate data, the limit of simulating up to one code per utterance can be very significant. In the set of six datasets it ranges from 65.95% - 98.04% of utterances being able to be replicated exactly. In the former's case, a model that can only simulate a single code per utterance would not be able to simulate about one third of possible utterances accurately.

Figure 5 shows that the average dataset contains 10.72 codes, for the 6 datasets with full access, it was found that to generate 95% of possible lines an average of 2.16 codes per line was required and to generate 99% of possible lines an average of 3 codes was required.

These two figures illustrate that the number of codes that appear in each utterance on average varies significantly. Additionally, the number of codes overall can vary from 6 to 15 The next section will discuss the model introduced in Swiecki et al. (2022) and evaluate it, looking at its limitations which can be used to find a gap in the research which can be addressed with a research project.

Number of Codes per Utterance	Average distribution of six datasets	Dataset the most skewed towards 0 and 1 codes	Dataset the least skewed towards 0 and 1 codes
0	58.957%	75.574%	51.971%
1	25.914%	22.299%	13.978%
2	11.849%	0.159%	34.050%
3	2.097%	1.112%	0.000%
4	0.969%	0.424%	0.000%
5	0.159%	0.212%	0.000%
6	0.014%	0.053%	0.000%
7+	0.041%	0.000%	0.000%

Fig 4. Distribution of codes in each utterance, One was the data used by Swiecki et al. (2022) and the other five were used data supplied by the International Society for Quantitative Ethnography. (n.d.)

Data Set	Total Number of Codes	What is the maximum number of codes that need to be produced in an utterance to generate 95% of lines in a dataset?	What is the maximum number of codes that need to be produced in an utterance to generate 99% of lines in a dataset?
Swiecki et al. (2022)	8	2	2
International Society for Quantitative Ethnography. (n.d.) Game of Thrones	12	2	3
International Society for Quantitative Ethnography. (n.d.) Debates	12	3	4
International Society for Quantitative Ethnography. (n.d.) Nephrotex	12	2	2
International Society for Quantitative Ethnography. (n.d.) Hamlet	10	1	3
International Society for Quantitative Ethnography. (n.d.) Romeo	10	3	4
12 Datasets minimum 6 maximum 15 see appendix for details	10.75	Dataset not Publically Available	Dataset not Publically Available
Average	10.72	2.16	3

Fig 5. A table displaying the number of codes in each dataset, those with full utterance data available, have been analysed to calculate the maximum number of codes that need to be simulated in a single utterance in order to be able to generate 95% and 99% of lines in the dataset.

3. Summary of the State of the Art

The literature review demonstrated that coded data is widely used to analyse collaborative discourse and showed that there are various methods of simulating collaborative discourse data. Additionally it demonstrated that coded datasets generally have multiple codes for each utterance. The most common type of collaborative discourse data which is being analysed uses explicit codes and content analysis is performed upon the datasets. There exists a single method found for simulating this specific type of discourse which is the model of Swiecki et al. (2022)

There are few limitations I have identified with this model when published:

1. The process only generates codes that appear in the data as opposed to an actual discourse (text or speech from a conversation).

2. The moving window only looks at the previous utterance, therefore the model may account for co-occurrences between codes that occur a greater number of utterances in the past.

3. The process only generates a maximum of 1 code per utterance. While most datasets contain utterances which contain multiple codes

Limitations 1 and 2 seem like they would be significant areas of improvement for the model. However, not generating the full conversations text makes the model multipurpose and does not require a large set of training data to adapt to the trends in the data. Generating whole utterances is already done in the model of Park et al., (2023). Furthermore if a full conversation was created, the data would need to be translated into coded data. This is a process that can be up to interpretation to decide if a specific utterance should be given a code. Limitation 2 has been addressed in an unpublished manuscript (Fang & Swieck., In preparation), solving the issue of a small moving window in the simulation.

Limitation 3 of being able to generate a maximum of 1 code per utterance is a significant gap that could be addressed in a research project, where the effect of the extension can be explored. There are two distinct benefits of this extension.

As shown in the literature review in figure 4 the range of utterances which can be simulated with up to one code per utterance is 65.95% - 98.04%. Attempting to use the current simulation on datasets at the lower end would likely create simulated discourse which is highly different to the input data as over 1/3rd of observed utterances could not be simulated. For these datasets where a significant proportion of utterances contain multiple codes the current model would likely be inadequate.

In addition, this extension would allow the model to simulate coded data that structural analysis is usually performed upon. Andrist et al. (2015) used codes such as “instructor gazing at reference ingredient”, “worker gazing at non reference ingredient”, “worker gazing at a different object than instructor” and “worker gazing at a same object than instructor”. Simulating codes for this dataset would requires multiple codes per line and this extension would therefore expand the types of coded data the model would be suitable to simulate.

4. Research Project Plan

Introduction

The project will involve extending the model proposed by Swiecki et al. (2022), which I have been given access to its full code base and utterance data which it was tested with. I will extend its functionality to allow it to produce data that can have more than one code in each utterance. The current implementation limits how close the simulated data can replicate that found in the real world.

The extension of the model would require a solution for two major areas. Firstly, what transition matrices to design, their number and how they would interact with each other. Secondly, by what exact process is the original coded data transformed into this new set of matrices.

Extensions to allow multiple codes

The model I will be extending upon creates two types of matrices. Firstly a singular speaker matrix that looks at who spoke last to calculate who should speak next. Secondly there is a code matrix for each speaker, the matrix for the chosen speaker is used to calculate what code they should speak.

We have considered two preliminary approaches to extending the model. The first of which is a naive approach that while addressing the limitation with the model directly, if dealing with a dataset with a large number of codes, it will create matrices likely too large for a consumer grade machine to store and perform operations upon. The iterative method seeks to address this foreseen issue.

The naive approach would be to expand the code matrix for each speaker instead of having 1 column per code, it would have every combination of codes, the limitation of such an approach is that each speaker's code matrix would be of size (n!,n!) where n is the number of codes in the dataset as compared to (n+1,n+1) in the current implementation. This idea could be developed further to try and minimise the size of the matrix perhaps by removing code combinations that are never seen in the input dataset. This approach uses a single matrix per speaker like with the original data but its dimensions are made larger.

The iterative method keeps the speaker's code matrix the same size, but has multiple matrices per speaker and repeats this process. More specifically, for each agent a distribution of the number of codes they discuss in each utterance is tracked as well as a set of matrices. The first matrix would be the same as in the original model, but there would be additional matrices to assign additional codes to a base utterance. Figure 6 compares these two approaches in more detail.

Both of these methods can be implemented and compared. The naive method would likely map closer to the conversation dynamics of the real data. However it may not be computationally feasible on most consumer devices. If attempting to simulate a dataset with a large number of possible codes. This would exceed both the indexing of most array implementations as well as the bytes available to store such a large object in working memory. The Iterative method in comparison should be quite close to the native method in the conversation dynamics generated. Moreover, all the matrices created are no larger than those used in the current method, there are just a greater number of them.

For example, if we compare a dataset of 4 codes and 5 speakers with one with 15 codes and 5 speakers. The current method would create a total of 6 arrays with 150 cells and 6 arrays with 1305 cells respectively. The naive method would create also 6 arrays but with 72,025 cells and 6 arrays with up to 2.18*10^27 cells. Meanwhile the iterative method would create 26 arrays with 545 cells and 81 arrays with up to 19,300 cells. This shows how the size of the arrays needed to store data for the naive approach is not workable for simulating a dataset with a large amount of codes.

Step	Steps of Current Method	Steps of Naive Method	Steps of Iterative Method
1	Input: 1. One speaker matrix of size (S,S) 2. For each speaker a matrix of size (C,C)	Input: 1. One speaker matrix of size (S,S) 2. For each speaker a matrix of size (C!,C!)	Input: 1. One speaker matrix of size (S,S) 2. For each speaker a set of up to N matrices where N is the maximum number of codes that each speaker uses in a single line. Each matrix is of size (C,C) 3. For each speaker, a matrix specifying the number of codes to generate of size (1,C)
2	Given X speaker spoke last which speaker should speak next	Given X speaker spoke last which speaker should speak next	Given X speaker spoke last which speaker should speak next, using input matrix 1
3	Given X spoke last and said Y code, what code should speaker Z say	Given X spoke last and said Y code combination, what code combination should speaker Z say	Given X spoke last and said Y code in their last utterance (pick at random if more than one code in the previous utterance), using input matrix 2
4	The process is repeated and each run generates a new utterance in the coded data	The process is repeated and each run generates a new utterance in the coded data	If the speaker used a code Calculate how many codes should be generated using input matrix 3
5			Columns that represent codes already in the current utterance need to be set to 0, as well as the null column, probabilities in the matrix recalculated so that each row totals 1 again. Repeat step 2 the number of times calculated,each time on a new matrix
6			The process is repeated and each run generates a new utterance in the coded data

Fig 6. Table comparing the various methods, “S” is the number of speakers, “C” is the number of codes plus 1

Process to create transition matrices

Each of these approaches also requires an algorithm in order to convert input data of coded utterances into the required matrices to build the model. However, this process is deterministic as compared to the simulation of data which is stochastic. An algorithm can be devised to convert an input coded dataset into transition matrices needed for each model. This process is already implemented for the current method and can be extended to create matrices for both the naive and iterative method.

Validation

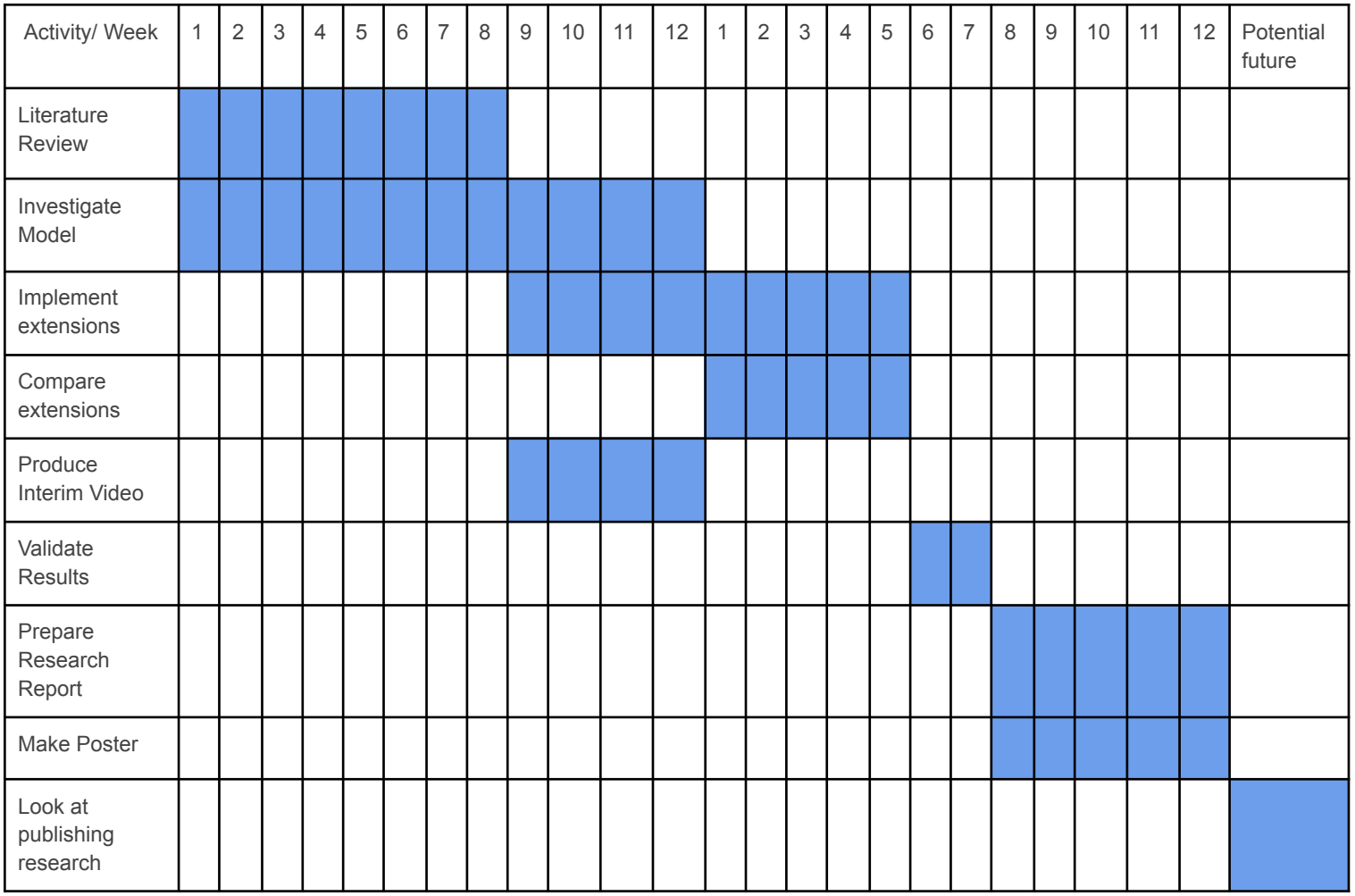
Validation of the simulation can be done in the same manner as the current model is validated which was developed by Swiecki (2021). This was done by testing a distribution of adjacency vectors on each simulated speaker and comparing it to the observed data, seeing if it is smaller than a 95th bias-corrected and accelerated percentile value. If it is smaller, it would suggest the simulated data is sufficiently similar to that of the input real world data. If an input array was given which only contained lines where there was a maximum of one line per utterance it would be expected that all models would perform similarly. On datasets with many utterances containing multiple codes, it would be expected that the extended models would perform significantly better

Report and Future Work

After the experiments are conducted a report will be written to discuss in detail what methods of extending the model were implemented, the results that were obtained and what was found during validation. With approval of Zachari Swiecki the author of the original model, the extended and documented model could be released on GitHub as open-source software and if time allows a relevant conference such as the International Conference on Computer Supported Collaborative Learning (CSCL) could be contacted to see if they would be interested in publishing the research.

Timeline of research project

Year	Week	Tasks
2023	1-8	• Complete a literature review and research project plan • Decide upon possible extensions to the existing model and the methods to do so • Investigate how the existing model works
2023	9-End of 2023	• Further understand how the existing model works and start at implementing the methods designed • Optional: Come up with and implement new methods that are conceived of after writing the project • Produce an interim video project presentation
2024	Start of 2024 - 5	• Finish implementing the various methods designed • Compare the implemented methods to investigate how suitable they are for varying numbers of codes and speakers when running the simulations on a single consumer grade high-end machine.
2024	6-7	• Validate and verify the simulated data
2024	8-12	• Write up research report • Prepare poster detailing my research
2024	12 - Potential Future	• Refine research report into a shorter journal article and send to a • relevant conference or journal



5. Conclusion

This report consisted of three major sections. Firstly, a literature review which investigated the types of analysis which is performed on collaborative data and the various methods of simulating this sort of data with agent-based models. It was found that most analysis of collaborative discourse data makes use of a coded dataset. The literature review followed with a focus on coded datasets exploring the distribution and number of codes that appear in the datasets. Secondly, a gap in the literature was found. It was identified that the model of Swiecki et al. (2022) simulated the type of data which is most often used in collaborative discourse research. A selection of limitations with this model were found and evaluated. The limitation of the model only being able to generate a maximum of one code per utterance was found to be one which could be feasibly addressed in a research project. Thirdly, this research project was laid out in detail, discussing, and comparing two preliminary concepts for an extension of the model, both of which in the project would be implemented and compared. The results of both can be compared with the original model on a variety of datasets and can be validated using the same methods that were used on the original model. The research project plan ended with a timeline detailing the progression of the project and the deliverables that need to be produced for the project.

6. Appendix

Data Set	Total Number of Codes
Swiecki et al. (2020)	9
Paquette et al. (2021)	15
Prieto et al. (2021)	13
Scianna et al. (2021)	8
Phillips et al. (2021)	12
Vega et al. (2021)	6
Jackowska (2021)	15
Siebert-Evenstone et al. (2021)	8
Espino et al. (2021)	14
Misiejuk et al. (2021)	6
Larson et al. (2021)	15
Schnaider et al. (2021)	8
Average	10.75

Fig 7. Coded datasets which did not have data publicly available

7. Reference List

Andrist, S., Collier, W., Gleicher, M., Mutlu, B., & Shaffer, D. (2015). Look together: Analyzing gaze coordination with epistemic network analysis. Frontiers in Psychology, 6. https://doi.org/10.3389/fpsyg.2015.01016

Brohinsky, J., Marquart, C., Wang, J., Ruis, A. R., & Shaffer, D. W. (2021). Trajectories in epistemic network analysis. Communications in Computer and Information Science, 106–121. https://doi.org/10.1007/978-3-030-67788-6_8

Boulay, D. B., Mitrovic, A., & Yacef, K. (2023). Handbook of Artificial Intelligence in Education. Edward Elgar Publishing.

De Marchi, S., & Page, S. E. (2014). Agent-based models. Annual Review of political science, 17, 1-20.

Espino, D. P., Wright, T., Brown, V. M., Mbasu, Z., Sweeney, M., & Lee, S. B. (2021). Student emotions in the shift to online learning during the COVID-19 pandemic. Communications in Computer and Information Science, 334–347. https://doi.org/10.1007/978-3-030-67788-6_23

Fang, Z. & Swiecki, Z. (In preparation). Simulating Collaborative Discourse Data.

Griffin, P., McGaw, B., & Care, E. (Eds.). (2018). Assessment and teaching of 21st Century skills: Research and applications. Springer.

Hamill, L., & Gilbert, N. (2010). Simulating large social networks in agent-based models: A social circle model. Emergence: Complexity and Organization, 12(4), 78-94.

Hutchins, E. (1996). Cognition in the wild. MIT Press.

International Society for Quantitative Ethnography. (n.d.). Coded datasets. International Society for Quantitative Ethnography. https://www.qesoc.org/coded-datasets/

Jackowska, M. (2021). How is team membership change manifested in collective interactions? – using epistemic network analysis to explore the challenges of contemporary team composition. Communications in Computer and Information Science, 292–303. https://doi.org/10.1007/978-3-030-67788-6_20

Kapur, M., Voiklis, J., & Kinzer, C. K. (2010). A complexity-grounded model for the emergence of convergence in CSCL groups. Analyzing Interactions in CSCL, 3–23. https://doi.org/10.1007/978-1-4419-7710-6_1

Katz, S., Aronis, J., & Creitz, C. (2000). Modeling pedagogical interactions with machine learning. Kognitionswissenschaft, 9(1), 45-49.

Koponen, I. T., & Nousiainen, M. (2018). An agent-based model of discourse pattern formation in small groups of competing and cooperating members. Journal of Artificial Societies and Social Simulation, 21(2). https://doi.org/10.18564/jasss.3648

Larson, S., Popov, V., Ali, A. M., Ramanathan, P., & Jung, S. (2021). Healthcare professionals' perceptions of telehealth: Analysis of tweets from pre- and during the COVID-19 pandemic. Communications in Computer and Information Science, 390–405. https://doi.org/10.1007/978-3-030-67788-6_27

Misiejuk, K., Scianna, J., Kaliisa, R., Vachuska, K., & Shaffer, D. W. (2021). Incorporating sentiment analysis with epistemic network analysis to enhance discourse analysis of Twitter data. Communications in Computer and Information Science, 375–389. https://doi.org/10.1007/978-3-030-67788-6_26

Oner, D. (2020). A virtual internship for developing Technological Pedagogical Content Knowledge. Australasian Journal of Educational Technology. https://doi.org/10.14742/ajet.5192

Paquette, L., Grant, T., Zhang, Y., Biswas, G., & Baker, R. (2021). Using epistemic networks to analyze self-regulated learning in an open-ended problem-solving environment. Communications in Computer and Information Science, 185–201. https://doi.org/10.1007/978-3-030-67788-6_13

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. arXiv preprint arXiv:2304.03442.

Phillips, M., Siebert-Evenstone, A., Kessler, A., Gasevic, D., & Shaffer, D. W. (2021). Professional decision making: Reframing teachers' work using epistemic frame theory. Communications in Computer and Information Science, 265–276. https://doi.org/10.1007/978-3-030-67788-6_18

Prieto, L. P., Rodríguez-Triana, M. J., Ley, T., & Eagan, B. (2021). The value of epistemic network analysis in single-case learning analytics: A case study in lifelong learning. Communications in Computer and Information Science, 202–217. https://doi.org/10.1007/978-3-030-67788-6_14

Schnaider, K., Schiavetto, S., Meier, F., Wasson, B., Allsopp, B. B., & Spikol, D. (2021). Governmental response to the COVID-19 pandemic - a quantitative ethnographic comparison of public health authorities' communication in Denmark, Norway, and Sweden. Communications in Computer and Information Science, 406–421. https://doi.org/10.1007/978-3-030-67788-6_28

Scianna, J., Gagnon, D., & Knowles, B. (2021). Counting the game: Visualizing changes in play by incorporating game events. Communications in Computer and Information Science, 218–231. https://doi.org/10.1007/978-3-030-67788-6_15

Siebert-Evenstone, A. L., Irgens, G. A., Collier, W., Swiecki, Z., Ruis, A. R., & Shaffer, D. W. (2017). In search of conversational grain size: Modeling semantic structure using moving stanza windows. Journal of Learning Analytics, 4(3), 123-139.

Siebert-Evenstone, A., Michaelis, J. E., Shaffer, D. W., & Mutlu, B. (2021). Safety first: Developing a model of expertise in collaborative robotics. Communications in Computer and Information Science, 304–318. https://doi.org/10.1007/978-3-030-67788-6_21

Shaffer, D. W., Collier, W., & Ruis, A. R. (2016). A tutorial on epistemic network analysis: Analyzing the structure of connections in cognitive, social, and Interaction Data. Journal of Learning Analytics, 3(3), 9–45. https://doi.org/10.18608/jla.2016.33.3

Shaffer, D. W., & Ruis, A. R. (2021). How we code. Communications in Computer and Information Science, 62–77. https://doi.org/10.1007/978-3-030-67788-6_5

Shah, M., Barany, A., & Siebert-Evenstone, A. (2021). "what would happen if humans disappeared from Earth?" tracing and visualizing change in a pre-school child's domain-related curiosities. Communications in Computer and Information Science, 232–247. https://doi.org/10.1007/978-3-030-67788-6_16

Swiecki, Z., Ruis, A. R., Farrell, C., & Shaffer, D. W. (2020). Assessing individual contributions to collaborative problem solving: A network analysis approach. Computers in Human Behavior, 104, 105876. https://doi.org/10.1016/j.chb.2019.01.009

Swiecki, Z. (2021) 'Measuring the impact of interdependence on individuals during collaborative problem-solving', Journal of Learning Analytics, 8(1), pp. 75–94. doi:10.18608/jla.2021.7240.

Swiecki, Z., Marquart, C., & Eagan, B. (2022). Simulating collaborative discourse data. In A. Weinberger, W. Chen, D. Hernandez-Leo, & B. Chen (Eds.), CSCL Proceedings - 15th International Conference on Computer-Supported Collaborative Learning (CSCL) 2022 (pp. 83-90). (Proceedings of International Conference of the Learning Sciences, ICLS). International Society of the Learning Sciences.

Swiecki, Z. (2022). The expected value test: a new statistical warrant for theoretical saturation. In B. Wasson, & S. Zörgő (Eds.), Third International Conference, ICQE 2021 Virtual Event, November 6–11, 2021 Proceedings (pp.

49-65). (Communications in Computer and Information Science; Vol. 1522). Springer. https://doi.org/10.1007/978-3-030-93859-8_4

Vega, H., Irgens, G. A., & Bailey, C. (2021). Negotiating tensions: A study of pre-service English as foreign language teachers' sense of identity within their community of Practice. Communications in Computer and Information Science, 277–291. https://doi.org/10.1007/978-3-030-67788-6_19

Yasnitsky, Anton. (2018). Vygotsky (1st ed.). Page 116. Routledge. https://doi.org/10.4324/9781315751504

Wooldridge, A. R., Carayon, P., Shaffer, D. W., & Eagan, B. (2018). Quantifying the qualitative with epistemic network analysis: A human factors case study of task-allocation communication in a primary care team. IISE Transactions on Healthcare Systems Engineering, 8(1), 72–82. https://doi.org/10.1080/24725579.2017.1418769

Extending a Model for Simulating Collaborative Discourse

Robert Brian Milligan
rmil0017@student.monash.edu
Monash University
Melbourne, Victoria, Australia

ABSTRACT

Collaborative discourse data, such as labelled transcripts of talk, is used to investigate collaboration in fields such as education, healthcare, psychology, and conflict resolution. However, obtaining these datasets is difficult for reasons such as privacy, a lack of applicable relevant data, and the cost and time commitments for collecting such data. Simulations for generating collaborative discourse data have been devised, but they fail to replicate important features of real data. In this paper, I extend a prior simulation method to explore two important features of collaboration—uptake, or how ideas overlap between consecutive discourse moves—and communication density, or the amount of relevant information in discourse moves. My results suggest that (1) even very small levels of uptake can have a relatively large and significant impact on models of collaborative discourse (2) the original simulation method is not appropriate for generating dense collaborative discourse; and (3) an extension the original method can simulate data that matches desired levels of density and replicates conversational dynamics. These results will help to guide researcher modelling decisions and provide a more valid approach to simulating collaborative discourse data.

CCS CONCEPTS

• **Human-centered computing → Collaborative and social computing**.

KEYWORDS

collaborative problem-solving, simulation, computational modelling, epistemic network analysis

1 INTRODUCTION

Collaborative learning is a form of learning that cultivates engagement, participation, critical thinking and problem solving skills [10]. It seeks to understand how groups work together to achieve a common goal be it sharing information, learning or solving problems. It is studied by researchers from a diverse range of academic disciplines and is used in a variety of domains including education[23], healthcare [12] and the military [21].

Collaborative discourse relates to how members of groups share ideas with one another and build upon each other's ideas. Obtaining real datasets to study the mechanics of collaborative discourse is often difficult for a variety of reasons including that often studies do not publish transcripts and coded datasets to protect the privacy of research participants. Moreover, those sources which are available are not always applicable to other scenarios and collecting new original data is often costly. Simulations can help address these issues by aiding social scientists and other researchers in designing experiments.

One of the more common forms of structuring collaborative discourse data is in the form of a coded dataset. This usually represents a group's conversation as a list of utterances over time said by specific speakers and sharing specific ideas or themes in these utterances which are called codes. These coded datasets are analysed using a variety of techniques including Epistemic Network Analysis (ENA) and its extensions such as Ordered Network Analysis (ONA) which model the relationships between codes and/or speakers by identifying relevant co-occurrences of codes. Such techniques can be used to compare the differences between coded datasets in order to understand the nature of collaboration in various scenarios

This paper investigates an extension of a model [20] to simulate coded datasets. This produces coded datasets that successfully model some interaction process measures, as described in more detail below, but is limited in its capability to model other important features of collaboration, such as uptake—how collaborators respond with similar ideas—and communication density—how much information is in a given utterance.

I propose and test an extension of the prior model that explicitly accounts for uptake and communication density. My results suggest that uptake is a critical feature of collaboration that can dramatically impact modelling decisions; moreover, I show that the updated model can accurately reproduce existing coded datasets. These results improve our understanding of collaboration and move the field closer to being able to more completely simulate relevant features of collaboration for research purposes.

2 BACKGROUND

In fields like the learning sciences and learning analytics investigating collaboration often involves collecting discourse data. One of the common forms of discourse data is a coded dataset. These contain conversations made up of a number of utterances. Each of these utterances is performed by a speaker in a group and may be labelled for one or more codes which can capture some meaning of the utterance, for example the content or themes present in the statement. Although the terminology conversation, utterance and speaker are used, a coded dataset can also describe non verbal communication such as written communication [5] or a record of observations made [19], or something separate such as readings from a sensor which tracked what object a participant was looking at [2].

Coded datasets can be analysed with a variety of methods; however Epistemic Network Analysis (ENA) [14] and Ordered Network Analysis (ONA) [22]have become popular techniques due to their ability to more validly model collaborative interactions by focusing on the relationships between coded concepts, rather than their presence in isolation [21, 7].

,

,

Useful results have come from such approaches, however collecting real discourse data is difficult for multiple reasons. Foremost respecting research participants' privacy limits the sharing of created coded datasets. As a result many publicly accessible coded datasets come from publicly available sources such as transcripts of debates and scripts from plays and tv shows [9]. Moreover, the conversations in existing datasets are not always applicable to other scenarios and conducting experiments to create new coded datasets is often time consuming and expensive.

To address these issues some researchers have used computational simulations [1, 3] and have used approaches such as agent based modelling which sets up specific behaviours for agents to follow and to generate datasets; however these datasets did not take the form of coded datasets, which are often used in studies of collaboration Simulating coded datasets has been proposed by Swiecki et al. [20] who designed and validated a method for simulating collaborative discourse data. While the method was useful it was limited in two important ways.

Firstly it was only used to investigate two features of collaboration, social and behavioural symmetry—i.e., the similarities and differences in who talks to whom and what they tend to talk about—but did not explore other potentially important features of collaboration, limiting its usefulness in practice. Secondly it only accounts for monothetic coding schemes, which assign one and only one code to each utterance. In practice, speakers may produce utterances that express multiple important concepts or ideas[11]. Thus, researchers often need to apply polythetic coding schemes to their data [16], which can assign multiple codes to a single utterance.

To address the first limitation, I used the existing model to investigate the notion of uptake. Uptake in collaborative discourse refers to a "non-accidental relationship between contributions" [17] and builds upon ideas of common ground as explored by Clark [6].Uptake is important as through the act of building on one another's ideas it shows understanding and promotes coherence within conversations. [13] Originally uptake was applied to transcripts of collaborative discourse conversations but since has been used with coded datasets [24]. In relation to our coded data sets the type of uptake that is measured is lexical overlap which refers to common ideas and concepts being discussed [17].

To address the second limitation I extended the existing model to allow for polythetic codes schemes that account for the various information densities in discourse. In collaborative discourse, communication density refers to the amount of meaningful information provided in a contribution [8]. As the existing polythetic coding schemes suggest, having control over the density of communication in the simulation should produce more realistic and valid results. Moreover, many coded datasets are polythetic and the ability to generate them or replicate the dynamics of an existing polythetic dataset would be very useful for researchers. Specifically, this paper seeks to address the following three research questions:

(1) What level of uptake is important to consider in collaborative discourse?
(2) How sensitive is the original simulation to different levels of communication density?
(3) Can the original mode be extended to account for higher levels of communication density?

3 METHODS

3.1 RQ 1

3.1.1 Existing Simulation framework. The existing simulation method replicates a social network of a group deciding the order of speakers in a conversation and who is more likely to speak in response to whom. This is modelled as a lag-1 Markov process using a single transition matrix for each group termed the speaker matrix. Taking the second row of figure 1 as an example, it indicates that if previously speaker 2 spoke, there is a 10% chance speaker 1 responds, 70% speaker 2 speaks again and 20% chance that speaker 3 responds.

A lag-1 Markov process is also used to model the content of the utterances using a transition matrix for each speaker in the group, termed the code matrix. In these matrices, utterance content is represented as the probability of co-occurrence between codes in their data. Taking figure 2 as an example, if the previous speaker said code A and the speaker matrix indicated that speaker 3 responded, then there is a 50% chance speaker 3 uses A, a 10% chance they use B, a 30% they use C and a 10% chance their utterance contains no codes.

Speaker	1	2	3
1	0.5	0.2	0.3
2	0.1	0.7	0.2
3	0.6	0.2	0.2

Figure 1: An example of a Speaker matrix where the number of speakers *s* is 3.

The original simulation method works as follows: Let *s* be the number of speakers and *n* be the number of codes including a null code. Construct a single speaker matrix of size *s* x *s* and *s* number code matrices each of size *n* x *n*. Each cell contains a positive number in the range [0,1] and each row of every matrix sums to 1. Initially a random speaker is chosen and one of the codes is set to 1; this creates the first utterance of the synthetic conversation. To construct the following lines, the algorithm examines the previous line of the dataset and identifies the speaker. This determines the row of the speaker matrix that is used and that row is sampled as a discrete random variable; this identifies the speaker for the following line. The code matrix for this specific speaker is then referred to. The previous utterance's code is used to determine the row of the code matrix used by the new speaker and again a discrete random variable is sampled to determine the code used in the following line by the new speaker. Following this process creates an additional utterance and this can be repeated a desired number of times to create a complete conversation–or collection of utterances each with an associated speaker and code. An example of a simulated conversation is shown in Figure 3.

Code For Speaker 1	A	B	C	Null
A	0.2	0.6	0.1	0.1
B	0.5	0.1	0.2	0.1
C	0.2	0.1	0.5	0.2
Null	0.2	0.3	0.1	0.4

Code For Speaker 2	A	B	C	Null
A	0.7	0.1	0.1	0.1
B	0.6	0.2	0.1	0.1
C	0.2	0.4	0.2	0.2
Null	0.1	0.2	0.2	0.5

Code For Speaker 3	A	B	C	Null
A	0.5	0.1	0.3	0.1
B	0.6	0.1	0.2	0.1
C	0.1	0.1	0.6	0.2
Null	0.2	0.1	0.1	0.6

Figure 2: An example of a set of code matrices where the number of codes is 4 (including a null code).

3.1.2 Operationalising uptake. I define uptake for this study as a number between 0 and 1 which captures the level of repeated codes between utterances. If an entire conversation never has an utterance coded for the same code twice in a row then uptake would be 0; conversely if the entire conversation consists of the same code present in every utterance, uptake is 1. Note that in the code matrices for each speaker described above, the diagonal of the matrix indicates the probability of the same code being present in consecutive utterances. Thus, uptake was operationalised in this simulation by creating code matrices where the diagonal was set to the result of sample from the normal distribution with a mean of the desired uptake and a standard deviation of 0.0001. All other columns were set to normalised random values so that each row totalled to 1. In sum, this process afforded the simulation of conversations with a desired uptake level by varying the diagonals of the speakers' code matrices.

3.1.3 Comparative analysis. To examine the importance of uptake in collaborative discourse, I simulated conversations at different levels of uptake. For each level, I compared two models of the simulated data—one that accounted for uptake and one that did not. If no uptake is present in the data, then these models should produce

Speaker	A	B	C	D	E	F	G	H	null_c
1	0	1	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	1	0
2	1	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	0	0	1	0
5	0	0	0	0	0	0	0	1	0
4	0	1	0	0	0	0	0	0	0
5	0	1	0	0	0	0	0	0	0
5	0	1	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	1

Figure 3: An example of a simulated conversation between 5 speakers and 9 possible codes; A-H as well as a null code.

identical results; as uptake in the data increases, the results of these models should diverge. The level of uptake at which these two models significantly differ, indicates the level of uptake that should be accounted for if researchers suspect that uptake is important to consider in their analysis.

For each level of uptake, the two models I compared were an ENA model and an ONA model. Both ENA and ONA model collaborative discourse as networks of connections between codes—that is, they represent how concepts, actions, or ideas were related to one another in the discourse [22, 15]. Both models construct networks for each speaker that indicates the relative frequency of co-occurrence between codes in their discourse and the discourse of other speakers in their group. To construct the networks for each speaker, a window is moved over the group's utterances and a co-occurrence is counted if a code is present in the speaker's current utterance and any of the previous utterances in the window.

The salient difference between the approaches for the purpose of this study is that ENA considers connections between the codes in utterances without regard to self connections between codes–i.e., uptake. In contrast, ONA considers connections between the codes in utterances taking into account self connections.

Both methods perform a dimensional reduction on the collection of networks identified in the data. Typically, this is done via singular value decomposition. However, in this study, a dimensional reduction technique was used that projects the networks down to one dimension that maximises the variance between two subpopulations in the data. Here, the two subpopulations were networks for the speakers that did not account for uptake (ENA networks) and networks for the same speakers that did account for uptake (ONA networks). That is, the networks were the same, except that one set included self-connections and the other set did not. Thus,

the dimensional reduction produced a scale that could be used to directly compare the difference between networks that either did or did not account for uptake. For more details on the ENA/ONA algorithm and dimensional reduction procedures, see [4].

To identify networks, a window size of 2 utterances was chosen for both models to align with the simulation procedure, which only accounts for the previous utterance when selecting speakers and codes. To statistically compare networks, I performed a pair-wise t-test between the one-dimensional scores for the two groups for each simulated dataset. To assess the size of the difference between the models for each simulated dataset, I calculated the effect size measure Cohen's *d*.

For each level of uptake, I generated a simulated dataset—where a dataset is a collection of 100 conversations from different groups and each group is defined by a different speaker matrix and set of code matrices. Each conversation included 300 utterances from five speakers with one of nine possible codes present in each utterance. The ENA and ONA models described above were compared on the dataset, producing an effect size and a p-value. This process was repeated 100 times, generating a distribution of effect sizes and p-values for each uptake level. The presence of uptake was considered significant when the lower bound of 95% confidence interval for the effect size distributions was consistently above 0.3 (a medium effect) and the upper bound of the 95% confidence interval for the p-value distributions was consistently below 0.05.

3.2 RQ 2

3.2.1 Operationalising communication density. For this study, I operationalised communication density as a value between 0 and 1. This is calculated as the number of codes present in each utterance of a conversation divided by the maximum number of codes that could appear in the entire conversation. If the entire conversation has no codes in each utterance the density is 0; conversely if every utterance contains every code then density is 1. This means a monothetic conversation's density is limited to 1 divided by the number of non-null codes in the dataset.

3.2.2 Validation. To explore the sensitivity of the original simulation framework to various levels of communication density, I used a collection of input datasets with known density levels, extracted the speaker and code matrices from these conversations, and used these matrices to generate simulated datasets. Next, I calculated the density level for the simulated datasets and compared them to the density levels measured from the input datasets. Finally, I compared ENA networks of the simulated datasets to the corresponding networks of the input datasets to assess the overall similarity of the simulated and input datasets. If the original simulation framework was sensitive to density, it should be able to produce (1) simulated datasets with density values that match the input datasets (on average) and (2) ENA networks of the simulated datasets that are highly similar to ENA networks of the input datasets.

I used a mixture of real-world and generated coded datasets as input datasets. I used a coded dataset [5] which had an average density of 0.095 and a modified version of this same dataset [20] which had an average density of 0.06. All other datasets were created by randomly generating lines where the probability of each code is the desired density[1]. Each of the input datasets consisted of a collection of 10 conversations varying in length between 111 and 279 utterances. Each conversation consisted of 5 speakers and in each utterance could be coded for 9 possible codes, including a null code. For each input dataset, I generated 1000 simulated datasets and compared the density of the input conversation to the average density of the simulated conversations.

To assess the overall similarity of the input and simulated datasets, I used the method proposed by Swiecki [18] to produce a measure that captures if the networks of the input and simulated datasets come from the same distribution. The average Euclidean distance of the input dataset's network is compared to the mean of the networks of the simulated datasets. Next, a null hypothesis distribution is generated by calculating the Euclidean distance of each simulated dataset's network to the mean of the simulated networks. If the former distance is less than the upper bound of the 95% confidence interval of the null hypothesis distribution, then the network of the input datasets and the networks of the simulated datasets—and by extension, the datasets themselves— are judged to be sufficiently similar.

3.3 RQ 3

3.3.1 Extended simulation framework. One naïve method for creating a model that can generate polythetically coded conversations is to increase the size of the code matrix to account for every combination of codes that could exist in an utterance. Like with the original method, let s be the number of speakers and n be the number of codes including a null code. A single speaker matrix of size s x s is created but in addition to this a series of s code matrices of size 2^n x 2^n are created. The process of using these matrices is the same manner as in the original method; however instead of identifying the previous code used in the previous utterance, the combination of codes used in the previous utterance is used to generate a combination of codes for the next utterance. Specifically, instead of identifying all individual codes in the previous and current utterances, a bitstring representing the previous and current utterance is used—e.g. for a conversation with 4 codes it would range from no codes in the utterance [0,0,0,0] to all codes in the utterance [1,1,1,1]. While this method can generate more than one code per utterance, it has a number of drawbacks including severe time and space complexity issues stemming from the large data structure required. For example, using the implementation of arrays in the Python package NumPy, to create a code matrix with 16 codes would require about 15 Gigabytes of memory and 24 codes would require 2 Petabytes of memory for each speaker's code matrix. When dealing with smaller datasets with fewer codes, it is possible to run simulations, create these matrices and generate conversations using consumer grade machines. However, the runtime for computing synthetic conversations was found to be extremely slow, taking 10 hours to produce 700 conversations of 111-279 utterances each of 9 possible codes. In comparison the original method took 6 minutes to produce the same number of synthetic conversations under the same conditions. Moreover, another limitation of this model is that it requires an increased processing overhead to convert a number of

[1]The desired density is not an exact density of the produced dataset but is very close e.g. dataset with a desired density of 0.1 was a dataset with a density of 0.1004079.

codes into a bitlist and vice versa. For these reasons the method was deemed unsuitable to replace the original model Another method for generating polythetically coded data —which I term the iterative method— utilises the two types of matrices found in the original method but also uses an additional matrix type. Like with the original method, let s be the number of speakers and n be the number of codes including a null code. A single speaker matrix of size s x s and s code matrices each of size n x n are constructed. However, an additional matrix —which I term the density matrix— is constructed for each speaker and is of size n. This holds a distribution of the number of codes the speaker uses in each utterance and can either be constructed from a dataset or from a given distribution such as a normal distribution with a mean number of codes to communicate.

Density	1	2	3	4
Speaker 1	0.2	0.5	0.3	0

Density	1	2	3	4
Speaker 2	0.1	0.8	0.1	0

Density	1	2	3	4
Speaker 3	0.3	0.4	0.3	0

Figure 4: An example of density matrices where the number of speakers s is 3, and the number of codes n is 4 (including a null code).

In the same manner as the other matrices, the density matrix as shown in Figure 4 contains a series of numbers in the range [0,1] and they sum to 1. To create a synthetic dataset with the iterative method, a random speaker is chosen and one of the codes is set to 1; this creates the first utterance of the synthetic conversation. Determining the next speaker is done in the same method as the original method. The density matrix is sampled to find the number of codes that should be in this speaker's next utterance. To choose the next utterance's codes,the codes in the previous utterance are considered. One of these previous codes is sampled uniformly and decides which row of code matrix is used in the next step. Next, this discrete random variable is sampled the corresponding number of times sampled from the density matrix to produce each code for the current utterance. Running this process creates a new utterance with 1 or more codes and this is used as the input for the next utterance for the synthetic dataset. This method is quite efficient and took 8 minutes to produce 700 conversations of 111-279 utterances each of 9 possible codes. Another benefit of this model is that by setting the density matrix for all speakers to [1,0,0...N] it has the exact same behaviour as the original model which produces 1 code per utterance.

3.3.2 Validation. The same process of validation used in research question two was applied to datasets generated by the iterative method.

4 RESULTS

4.1 RQ 1

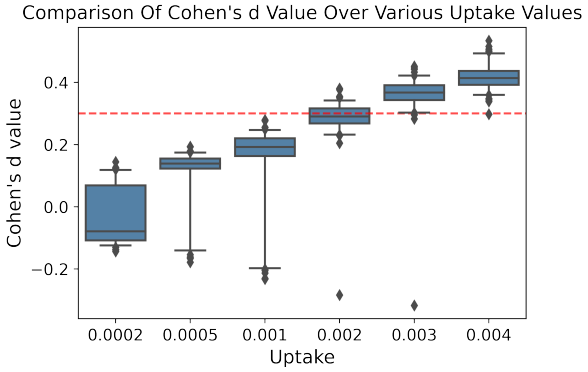


Figure 5: Effect size (Cohen's d) of difference between uptake and no uptake models for varying levels of uptake. The dashed line indicates a "medium" effect size.

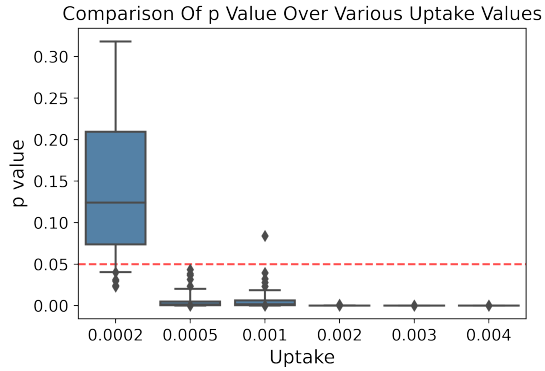


Figure 6: p value of difference between uptake and no uptake models for varying levels of uptake. The dashed line indicates a statistically significant result at $\alpha = 0.05$.

Figures 5 and 6 show the results of the comparison between the outputs of the uptake and no-uptake models on simulated datasets for varying levels of uptake. Figure 5 shows the differences in terms of effect size as measured by Cohen's d; figure 6 shows the associated p values. Preliminary analyses suggested that very small levels of uptake in the data lead to large and significant differences between the models. To more precisely identify just how much uptake made a difference, I used a fine-grained scale for the uptake values ranging from 0.0002 to 0.004. Overall, the results suggest that seemingly negligible amounts of uptake in the data can produce significant differences in the models. At an uptake level of 0.0005, differences between the models for all tests resulted in p values less than 0.05; however, the effect size of these differences remained relatively small. At an uptake level of 0.004, all p values were significant and the lower bound of the effect size distribution

,

was above a Cohen's d of 0.3, indicating that over 95% of the tests obtained effect sizes considered medium sized. This demonstrates that very small amounts of uptake produce relatively large and significant differences in conversation dynamics.

4.2 RQ 2

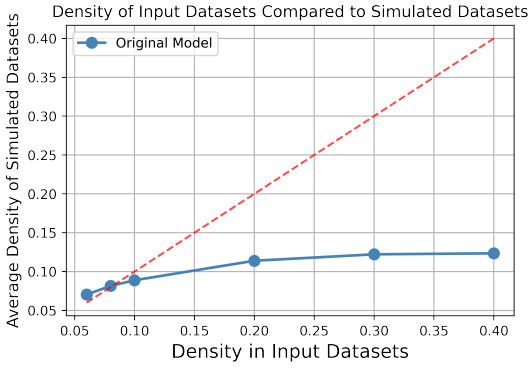


Figure 7: Comparison of density between an input dataset and the average density for 1000 simulated dataset using the original simulation framework. The dashed line indicates matching input and output densities—the ideal result.

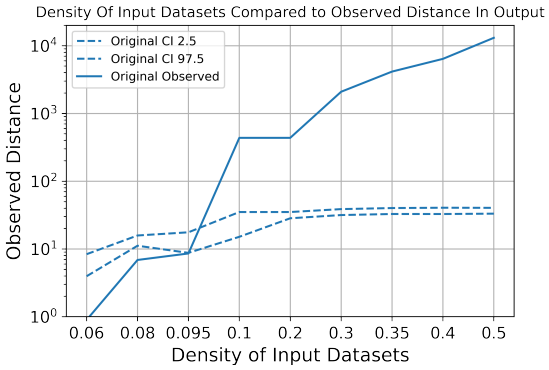


Figure 8: Comparison of an observed distance between an input dataset and 1000 simulated datasets for a variety of densities. The dashed lines indicate a 95% confidence interval.

Figures 7 and 8 show the performance of the original simulation method over a series of increasing densities of input datasets. Figure 7 compares the density of the input datasets to the average of the simulated dataset. Figure 8 compares the overall similarity of the input datasets and simulated dataset in terms of networks generated from the data. Figure 7 shows that the original simulation framework is able to match density levels until a level of 0.1. However, it is unable to generate sufficiently dense datasets when the input data has a density above 0.1. As the original model can only generate monothetic conversations, the maximum density it can produce is 1 divided by the number of non null codes or 0.125 for this dataset. As seen in figure 8 the dynamics of the conversations begin to significantly differ around the same level, the distance between the network on the input dataset and simulated datasets falling higher than the upper bound of the 95% confidence interval. Increasing the density higher than this increases the discrepancy between input and output datasets density as well as the dissimilarity of conversation dynamics. This demonstrates the unsuitability of the original model for simulating polythetic datasets, not just in terms of the density of conversation generated but also with dynamics of it matching the input data source.

4.3 RQ 3

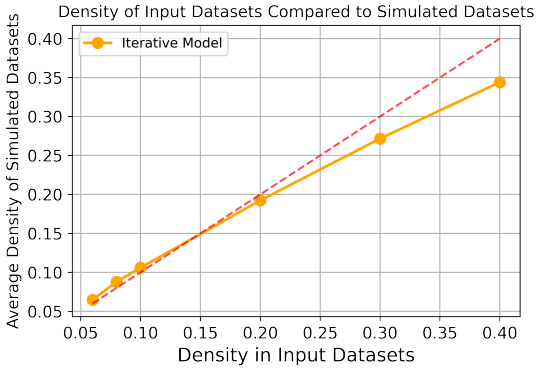


Figure 9: Comparison of density between an input dataset and the average density for 1000 simulated datasets using the iterative model. The dashed line indicates where the input and output densities match.

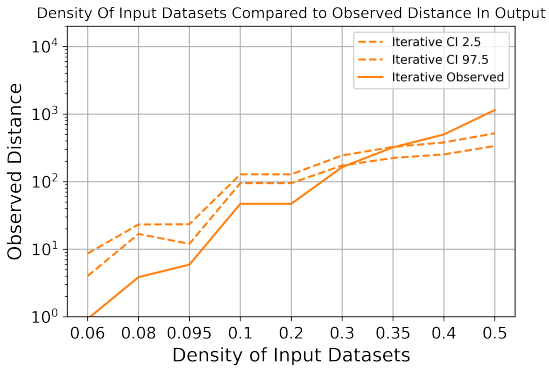


Figure 10: Comparison of an observed distance between an input dataset and 1000 simulated datasets for a variety of densities. The dashed lines indicate a 95% confidence interval.

Figures 9 and 10 show the performance of the iterative model over a series of increasing densities of input datasets. Figure 9 compares the density of the input dataset to the average of the simulated datasets. Figure 10 compares the overall similarity of the input datasets and simulated datasets in terms of networks

generated from the data. Figure 9 shows that the iterative model can handle producing datasets of a higher density. Its ability to do so however starts to drop off when the input density reaches 0.35-0.4 where the method can no longer generate sufficiently dense datasets, only generating datasets with approximately 86-90% of the density of the input data at that level. Additionally as seen in figure 10, at this point the datasets generated are no longer statistically identical, falling outside of the 95% confidence interval range. This demonstrates that the iterative model is suitable for simulating coded datasets for a variety of densities that represent polythetic datasets; while also ensuring the dynamics in them are similar to those found in the input coded dataset.

5 DISCUSSION AND CONCLUSIONS

The aims of the study were threefold: (1) to discover at what level is uptake significant in collaborative discourse; (2) to evaluate how sensitive the original simulation method is to conversation density; and (3) to extend the original simulation method to allow for datasets of higher densities—i.e., polythetically coded datasets. It was found that very small changes in uptake cause significant differences between models of collaborative discourse. Given a dataset of 9 codes, the original simulation method could replicate the dynamics and density of datasets below 0.1, but performed poorly for replicating both the density and similarity to higher densities datasets. In comparison, the iterative method could produce datasets of a much higher density and matched a similar degree of density and dynamics when the density of the input datasets was below 0.35. This means that iterative model can successfully produce a range of polythetic coded datasets This study had several limitations. First, the iterative model did not keep up with the density of the input datasets at very high density levels. This may be due to an assumption with the design of the iterative model which causes it to draw its generated codes with replacement. This is to account for cases where density matrix suggests more codes should be generated than the code matrix could generate; e.g. the density matrix suggests 4 codes should be in an utterance but the code matrix only has 3 non zero values for a particular row. The result of this is a limitation that the density of datasets generated by the iterative model does not match the input data for high densities. The density at which the method broke down corresponds to almost half of the total number of codes appearing in each utterance. Published studies do not report density levels, however, the real dataset [6] examined in this study only had a density of 0.095. So, it may be that such high densities are unlikely in real world scenarios, though further research is needed to test this claim. Second, the study used fixed parameters that could potentially impact the results. In particular, all simulated conversations used in the study could be coded for 9 possible codes. This is a reasonable amount as it was seen in the real data used to inform our simulations; however, real world datasets can vary widely in the number of codes used (see original literature review) and the results presented here may differ depending on how many different codes an utterance could contain. In addition a moving window of size 2 was used to consider if codes in a temporal space were related to each other. This was the choice made for generating matrices used in the original simulation method and was continued in this study. Increasing the moving window size would impact the construction of the code matrices and the generated conversations as a result. Future work will explore the interaction of these parameters with the outcomes investigated here. Despite these limitations, the findings are significant as they demonstrate that considering small levels of uptake can cause significant differences in models of collaborative discourse. This highlights the importance of choosing whether ENA or ONA is appropriate for analysis of specific coded datasets—if uptake is present in the data and research suspects that it is meaningful to model, ONA is the more valid option. In addition, it is useful for social scientists and collaboration researchers using the original simulation method to know that it is not appropriate for polythetic coded datasets. This paper introduces an extension of the original method which is more appropriate for producing and replicating many polythetic datasets as seen in the real world. The extension is quite versatile and performs well for quite dense conversations. The iterative model does not only produce denser conversations but also ensures the dynamics of the conversation are kept. This extended framework may be used by researchers to investigate collaboration using more diverse and realistic simulated datasets.

REFERENCES

[1] Jan Andersson and Jerker Rönnberg. 1995. Recall suffers from collaboration: joint recall effects of friendship and task complexity. *Applied Cognitive Psychology*, 9, (June 1, 1995), 199–211. DOI: 10.1002/acp.2350090303.

[2] Sean Andrist, Wesley Collier, Michael Gleicher, Bilge Mutlu, and David Shaffer. 2015. Look together: analyzing gaze coordination with epistemic network analysis. *Frontiers in Psychology*, 6. Place: Switzerland Publisher: Frontiers Media S.A. DOI: 10.3389/fpsyg.2015.01016.

[3] Yoav Bergner, Jessica Andrews-Todd, Mengxiao Zhu, and Joseph Gonzales. 2016. Agent-based modeling of collaborative problem solving. *ETS Research Report Series*, 2016, (July 1, 2016). DOI: 10.1002/ets2.12113.

[4] Dale Bowman, Zachari Swiecki, Zhiqiang Cai, Yeyu Wang, Brendan Eagan, Jeff Linderoth, and David Williamson Shaffer. 2021. The mathematical foundations of epistemic network analysis. In *Advances in Quantitative Ethnography : Second International Conference, ICQE 2020 Malibu, CA, USA, February 1–3, 2021 Proceedings*. International Conference on Quantitative Ethnography 2020. Springer, 91–105. DOI: 10.1007/978-3-030-67788-6_7.

[5] Naomi Chesler, Andrew Ruis, Wes Collier, Zachari Swiecki, Golnaz Arastoopour, and David Shaffer. 2014. A novel paradigm for engineering education: virtual internships with individualized mentoring and assessment of engineering thinking. *Journal of biomechanical engineering*, 137, (Nov. 1, 2014). DOI: 10.1115/1.4029235.

[6] 1996. Common ground. In *Using Language*. 'Using' Linguistic Books. Herbert H. Clark, (Ed.) Cambridge University Press, Cambridge, 92–122. ISBN: 978-0-521-56158-7. DOI: 10.1017/CBO9780511620539.005.

[7] Andras Csanadi, Brendan Eagan, Ingo Kollar, David Shaffer, and Frank Fischer. 2018. When coding-and-counting is not enough: using epistemic network analysis (ENA) to analyze verbal data in CSCL research. *International Journal of Computer-Supported Collaborative Learning*, 13, (Nov. 26, 2018). DOI: 10.1007/s11412-018-9292-z.

[8] Nia M. M. Dowell, Tristan M. Nixon, and Arthur C. Graesser. 2019. Group communication analysis: a computational linguistics approach for detecting sociocognitive roles in multiparty interactions. *Behavior Research Methods*, 51, 3, (June 1, 2019), 1007–1041. DOI: 10.3758/s13428-018-1102-z.

[9] International Society for Quantitative Ethnography. [n. d.] Coded datasets. Retrieved May 21, 2024 from https://www.qesoc.org/coded-datasets/.

[10] Anuradha A. Gokhale. 1995. Collaborative learning enhances critical thinking. *Journal of Technology Education*, 7, 1.

[11] Arthur Graesser, Natalie Person, and John Huber. 1993. Question asking during tutoring and in the design of educational software. In 149–172. ISBN: 978-1-315-04471-2. DOI: 10.4324/9781315044712-6.

[12] Rowan Myron, Catherine French, Paul Sullivan, Ganesh Sathyamoorthy, James Barlow, and Linda Pomeroy. 2018. Professionals learning together with patients: an exploratory study of a collaborative learning fellowship programme for healthcare improvement. *Journal of Interprofessional Care*, 32, 3, (May 2018), 257–265. DOI: 10.1080/13561820.2017.1392935.

[13] Martin Nystrand, Lawrence L. Wu, Adam Gamoran, Susie Zeiser, and Daniel A. Long. 2003. Questions in time: investigating the structure and dynamics of

unfolding classroom discourse. *Discourse Processes*, 35, 2, 135–198. Place: US Publisher: Lawrence Erlbaum. DOI: 10.1207/S15326950DP3502_3.

[14] David Shaffer, David Hatfield, Gina Svarovsky, Padraig Nash, Aran Nulty, Elizabeth Bagley, Kenneth Frank, André Rupp, and Robert Mislevy. 2009. Epistemic network analysis: a prototype for 21st-century assessment of learning. *International Journal of Learning and Media*, 1, (May 1, 2009), 33–53. DOI: 10.1162/ijlm.2009.0013.

[15] David Williamson Shaffer, Wesley Collier, and A. R. Ruis. 2016. A tutorial on epistemic network analysis: analyzing the structure of connections in cognitive, social, and interaction data. *Journal of Learning Analytics*, 3, 3, (Dec. 19, 2016), 9–45. Number: 3. DOI: 10.18608/jla.2016.33.3.

[16] Robert R. Sokal. 1974. Classification: purposes, principles, progress, prospects. *Science*, 185, 4157, (Sept. 27, 1974), 1115–1123. Publisher: American Association for the Advancement of Science. DOI: 10.1126/science.185.4157.1115.

[17] Daniel D. Suthers and Caterina Desiato. 2012. Exposing chat features through analysis of uptake between contributions. In *2012 45th Hawaii International Conference on System Sciences*. 2012 45th Hawaii International Conference on System Sciences (HICSS). IEEE, Maui, HI, USA, (Jan. 2012), 3368–3377. ISBN: 978-1-4577-1925-7 978-0-7695-4525-7. DOI: 10.1109/HICSS.2012.274.

[18] Zachari Swiecki. 2022. The expected value test: a new statistical warrant for theoretical saturation. In *Third International Conference, ICQE 2021 Virtual Event, November 6–11, 2021 Proceedings*. International Conference on Quantitative Ethnography 2021. Springer, 49–65. DOI: 10.1007/978-3-030-93859-8_4.

[19] Zachari Swiecki and Brendan Eagan. 2023. The role of data simulation in quantitative ethnography. In *Advances in Quantitative Ethnography - 4th International Conference, ICQE 2022 Copenhagen, Denmark, October 15–19, 2022 Proceedings*. International Conference on Quantitative Ethnography 2022. Springer, 87–100. DOI: 10.1007/978-3-031-31726-2_7.

[20] Zachari Swiecki, Cody Marquart, and Brendan Eagan. 2022. Simulating collaborative discourse data. In *CSCL Proceedings - 15th International Conference on Computer-Supported Collaborative Learning (CSCL) 2022*. International Conference on Computer-Supported Collaborative Learning 2022. International Society of the Learning Sciences, 83–90. Retrieved May 21, 2024 from https://research.monash.edu/en/publications/simulating-collaborative-discourse-data.

[21] Zachari Swiecki, A. R. Ruis, Cayley Farrell, and David Williamson Shaffer. 2020. Assessing individual contributions to collaborative problem solving: a network analysis approach. *Computers in Human Behavior*, 104. Place: Netherlands Publisher: Elsevier Science. DOI: 10.1016/j.chb.2019.01.009.

[22] Yuanru Tan, Andrew R. Ruis, Cody Marquart, Zhiqiang Cai, Mariah A. Knowles, and David Williamson Shaffer. 2023. Ordered network analysis. In *Advances in Quantitative Ethnography*. Crina Damşa and Amanda Barany, (Eds.) Springer Nature Switzerland, Cham, 101–116. ISBN: 978-3-031-31726-2. DOI: 10.1007/978-3-031-31726-2_8.

[23] Anouschka van Leeuwen and Jeroen Janssen. 2019. A systematic review of teacher guidance during collaborative learning in primary and secondary education. *Educational Research Review*, 27, (June 1, 2019), 71–89. DOI: 10.1016/j.edurev.2019.02.001.

[24] Yeyu Wang, Andrew Ruis, and David Shaffer. 2023. Modeling collaborative discourse with ENA using a probabilistic function. In (Apr. 29, 2023), 132–145. ISBN: 978-3-031-31725-5. DOI: 10.1007/978-3-031-31726-2_10.

Part 3: Appendices

The formatting of the research paper is consistent with the ACM conference proceedings which is the required format for the International Learning Analytics and Knowledge Conference (Q1).

Original model code base https://github.com/zlswiecki/cscl_2022_swiecki_marquart_eagan

My codebase https://github.com/robertmxmx/Swiecki_Extended_Model