Analysis of Language Change in Collaborative Instruction Following

Anna Effenberger[1], **Eva Yan**[2]*, **Rhia Singh**[2]*, **Alane Suhr**[1], and **Yoav Artzi**[1]
[1]Cornell University
[2]City University of New York
ae347@cornell.edu eyan0749@gmail.com
rhia.singh@macaulay.cuny.edu {suhr, yoav}@cs.cornell.edu

Abstract

We analyze language change over time in a collaborative, goal-oriented instructional task, where utility-maximizing participants form conventions and increase their expertise. Prior work studied such scenarios mostly in the context of reference games, and consistently found that language complexity is reduced along multiple dimensions, such as utterance length, as conventions are formed. In contrast, we find that, given the ability to increase instruction utility, instructors increase language complexity along these previously studied dimensions to better collaborate with increasingly skilled instruction followers.

1 Introduction

Community language change in situated collaborative task-oriented scenarios has been studied with focus on reference games (Krauss and Weinheimer, 1964; Clark and Wilkes-Gibbs, 1986; Hawkins et al., 2017, 2020a,b), where two participants coordinate using language to select to a single item from a set of available items. These studies found that utility-maximizing participants trade surface-form linguistic complexity with established norms, as the familiarity and expertise of the interaction partners increase. In practice, this emerges as a reduction in utterance length and vocabulary size.

We study the generality of these observations by analyzing language change in a collaborative instructional task, where instructors can specify multiple goals within a single instruction to increase their utility. This option, not present in reference games, creates competing incentives: increasing utility by issuing more goals in a single instruction versus decreasing language effort by utilizing established norms (e.g., by shortening instructions).

We use the CEREALBAR game environment and its accompanying dataset (Figure 1; Suhr et al., 2019). CEREALBAR is a two-player, collaborative language game where players work together to collect sets of matching cards. A leader plans which cards to include in the next set, and writes instructions to a follower describing tasks to accomplish. In contrast to reference games (Krauss and Weinheimer, 1964), the language in CEREALBAR is primarily instructional rather than referential, and the game allows players to complete a dynamic number of tasks per instruction and game.

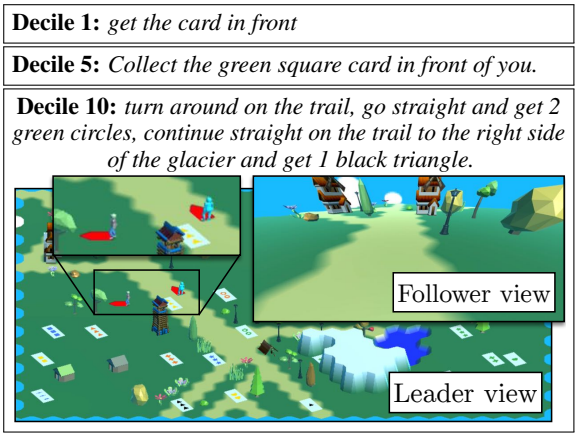

Figure 1: Leader instructions in CEREALBAR from games played at the beginning (Decile 1), middle (Decile 5), and end (Decile 10) of the community life. The differences between the instructions illustrate the linguistic change observed in the data. The instruction from Decile 10 is paired with a snapshot from the game as the follower begins to execute it. The leader (left) and follower (right) are highlighted in the center-left of the leader's view of the game, and the top right shows the follower's first-person view of the environment.

Similar to previous studies, we observe language change over time along the same dimensions. But, unlike in reference games, we observe utterance-level linguistic complexity increases. Our study illustrates that the formation of common ground among interaction participants does not necessarily reduce language complexity, and may even come with an increase in complexity. Understanding how humans use language to collaborate in settings with flexible utility is key to building natural

*Equal contribution.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 2803–2811
November 7–11, 2021.

	Mean	Median	Max
Interaction Score (# Card Sets)	8.8	10.0	19
# Instructions / Interaction	22.0	26.0	41
# Tokens / Instruction	14.4	13.0	55
Vocabulary Size		3,499	
Total # Instructions		17,524	

Table 1: Statistics of analyzed data.

language systems that effectively collaborate with users over time. Our analysis code can be found at github.com/lil-lab/CB-analysis.

2 Scenario and Data Overview

We use the CEREALBAR game and accompanying dataset (Suhr et al., 2019) in our analysis. CEREALBAR is a collaborative, two-player game, where a leader and a follower collect matching sets of cards by moving in an environment. The game is turn-based, and each player has a limited number of steps per turn. The leader both collects cards and instructs the follower using natural language.[1] The follower executes leader instructions. The players' abilities differ: the leader observes the complete environment and plans sets to collect; the follower only observes what is ahead, but has more steps per turn. For each set made, players receive one point and additional turns, allowing them to complete more sets. Success requires the players to collaborate via natural language: the leader must write informative instructions to the follower, and the follower must efficiently follow these instructions. Figure 1 shows a snapshot of the game.

The CEREALBAR dataset contains 1,202 human-human game interactions collected over the course of four months. Workers were randomly assigned as leader or follower for each interaction. The collection process created a Wizard-of-Oz setup: the system user, as the leader, provides instructions and acts in the world, and the human follower is a wizard, executing instructions to emulate the desired system behavior. We only use interactions from the training split for our analysis. We prune interactions by inexperienced workers, as classified when the data was collected, to focus on the impact of experience.[2] In total, we consider 795 interactions. Table 1 provides basic statistics of the data we use. Suhr et al. (2019) used these data to train models, while we study how the language changes.

[1] All utterances are in English.

[2] Appendix B.1 describes this pruning process.

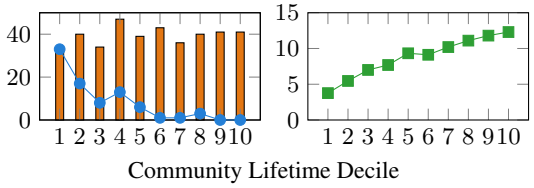

Figure 2: Community size (left) and mean game score (right) over deciles of community lifetime. On the left, the bars show total active players and the curve shows only the number of new players that joined per decile.

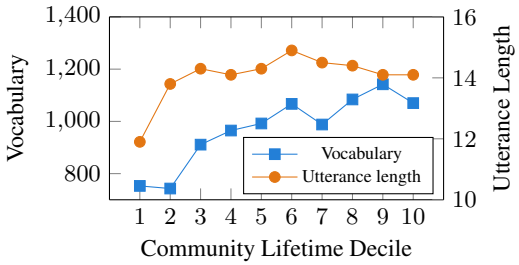

Figure 3: Vocabulary and utterance length over deciles.

3 Data Analysis

To analyze trends over the data collection period, we split the data chronologically into 10 deciles of roughly equal size (79 or 80 interactions). An average of 40 workers participated in each decile (Figure 2, left). The community stabilized after Decile 4, as worker recruitment slowed and the community was split by expertise.[3]

Interaction goals are increasingly achieved over time. Mean score per game increases from 3.8 to 12.3 ($p < 0.0001$) (Figure 2, right).[4] Execution efficiency and game expertise also improve.[5] Our focus is how leader language - the sole communication conduit - changes to enable these gains.

We design our analysis to be as similar as possible to existing work on reference games (Hawkins et al., 2020a), which shows that certain language aspects are simplified as community conventions form. CEREALBAR allows for a different realization of common ground development than previously studied reference games, and we observe trends that are in contrast to this line of prior work.

Instruction Length and Vocabulary Mean[6] instruction length increases from 11.9 to 14.1 tokens[7] ($p < 0.0001$) over time, while vocabulary size in-

[3] Appendix B.2 provides decile details.

[4] We use a two-sided t test at $\alpha = 0.05$ for all calculations of significance when comparing means.

[5] Appendix C.1 details this improvement.

[6] All means over instructions are first computed within each game, then across games. This weighs all games equally, rather than upweighing longer, higher-scoring games.

[7] We use NLTK for tokenization, lowercase all tokens, and use the `autocorrect` library for typo correction.

creases from 752 to 1,070 unique tokens (Figure 3). This contrasts with reference games, where utterance length and vocabulary size reduce (Clark and Wilkes-Gibbs, 1986; Hawkins et al., 2017). Some of the words added more specifically describe props or movements. However, the overall trend is relatively complex, and identifying clear patterns likely requires a more targeted scenario design.

Syntactic Complexity We analyze syntactic trends using parts-of-speech (POS) tags and dependency trees.[8] We do not observe a significant difference in usage of closed- and open-class POS tags, as seen in reference games (Hawkins et al., 2017). We observe change in the relative use of verbs, nouns, conjunctions, determiners, and numerals.[9] Notably, the proportion of conjunctions of all tokens increases from 0.060 to 0.067 ($p = 0.0026$).[10] The proportion of instructions that contain a conjunction also increases from 0.0495 to 0.0707 ($p = 0.0113$). Qualitatively, this accompanies an increased use of ordered sentential conjunctions, often to specify multiple tasks in a single utterance (e.g., *once you get that card, turn around and go left and get the 1 green circle card*).

We compute three measures of syntactic complexity using dependency trees (Xu and Reitter, 2016): (a) maximum depth: the longest path from root to a leaf; (b) maximum width: the maximum out-degree of any node; and (c) average branching factor: the average out-degree of non-leaf nodes.[11] We normalize all measures to control for utterance length. Figure 5 shows these statistics over time. Maximum width and branching factor increased from 0.941 to 0.987 ($p = 0.0483$) and from 0.934 to 1.00 ($p = 0.0051$), indicating increased descriptiveness. Maximum depth did not significantly change, indicating embedded clause use proportional to length, as expected when increasingly combining instructions with conjunctions.

We observe similar trends when measuring these statistics when comparing low- and high-scoring games (Figure 4). Higher scoring games had, on average, instructions with significantly higher width and branching factor. In Decile 1, language in games scoring 1 point and 16 points had an average normalized branch factor of 0.915 and 1.02. However, games in the lower 50% of scores showed a higher increase in syntactic complexity over time.

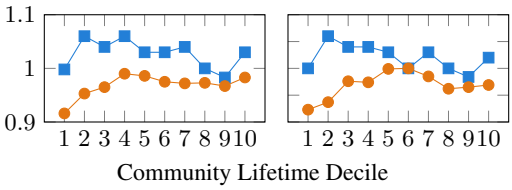

Figure 4: Average dependency branching factor (left) and maximum width (right) over deciles split to games that were above (blue) / below (orange) that decile's median game score.

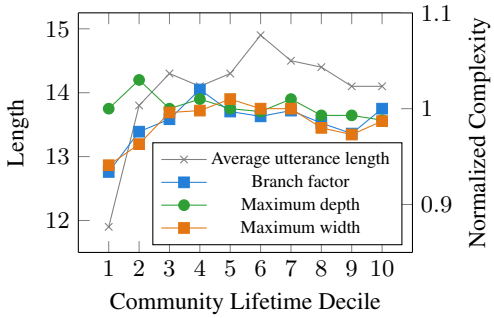

Figure 5: Average syntactic branching factor, maximum depth, and maximum width across deciles. We also plot the mean utterance length for reference.

Overall, our syntactic analysis shows an increase in language complexity is required to describe more tasks within a single instruction. We do not observe a gradual drop of redundant modifiers and descriptors (Hawkins et al., 2017). This may be because potential referents do not pose as much ambiguity as the abstract shapes often used in reference games (Clark and Wilkes-Gibbs, 1986).

Changes in References We see no significant development of niche idioms, in contrast to reference games with abstract shapes (Hawkins et al., 2020a). This is likely due to concreteness and familiarity of the referents in CEREALBAR, allowing players to rely on common background knowledge with little ambiguity. We observe change in the relative frequency of references to specific objects over time. We consider seven object classes: building, road, foliage, rock, ice, water, and light.[12] The proportion of instructions containing a reference to ice, light, and buildings increase from 0.006 to 0.022 ($p = 0.0006$), from 0.015 to 0.027 ($p = 0.0188$), and from 0.056 to 0.073 ($p = 0.0436$). The ratios of other references are stable. Leaders likely choose references to balance informativity and effort. Foliage objects are common and require

[8] We use spaCy (Honnibal and Montani, 2017) for POS tagging and dependency parsing.

[9] Appendix C.2 provides details.

[10] We use a one-sided z test at $\alpha = 0.05$ for calculations of significance when comparing proportions.

[11] We further explain the syntactic measures and provide example instructions for illustration in Appendix C.2.

[12] Appendix C.3 describes this classification process.

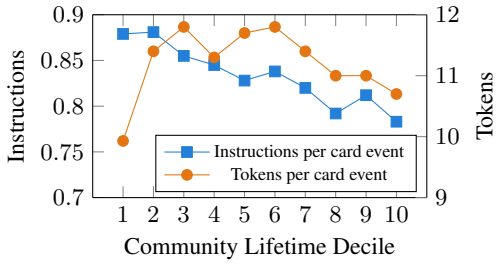

Figure 6: The number of instructions and tokens required for a card event over deciles. Analysis considers only instructions marked complete by the follower.

more effort to differentiate, while buildings and ice clearly vary. Lights, though common, were often referred to with other objects to clarify location.

Language Effort Leaders in CEREALBAR mainly instruct followers to complete card events to ultimately select valid card sets. We measure language effort with respect to this objective as the number of tokens and instructions per card event (Figure 6). This notion of effort is similar to utterance cost in speaker-listener pragmatic models (Goodman and Frank, 2016). The number of instructions per card event decreases from 0.879 to 0.783 ($p = 0.0102$), indicating leaders effectively pack more tasks into fewer instructions – often multiple card events into one instruction in later deciles (Figure 1). This change correlates with structural changes. For example, conjunctions are useful to pack more tasks into single instructions; the correlation across deciles between the proportion of instructions containing a conjunction and the number of instructions per card event is $r = -0.8243$. The high negative correlation indicates that the change in conjunction use aligns with the increase in goals (i.e., cards to select) packed per instruction. The number of tokens per card event initially increases from 9.9 to 11.8, then decreases to 10.7. This may be because, initially, followers require more verbose instructions and leaders experiment with the level of description, but as conventions form, this verbosity is less needed to understand instructions.

The reduction in the number of tokens per goal later on corresponds to the reduction in utterance length observed in reference games (Hawkins et al., 2017), although it is manifested differently as the overall surface-form is not simplified (i.e., via shorter utterances), unlike in reference games. Given the opportunity to increase utility, leaders choose to take advantage of followers' increased expertise and efficiency by using more complex language to pack more goals into each instruction.

4 Discussion and Related Work

The CEREALBAR scenario is related to reference games (Krauss and Weinheimer, 1964; Clark and Wilkes-Gibbs, 1986; Hawkins et al., 2017; Monroe et al., 2017; He et al., 2017; Udagawa and Aizawa, 2019; Haber et al., 2019), which require two players to agree on a single referent from a set via dialogue. CEREALBAR differs in several ways. It allows only unidirectional language communication, and utterances in CEREALBAR are instructions specifying desired follower behavior with any number of tasks to complete (i.e., with flexible utility), not a description of a single target referent.

These differences lead to different language dynamics. In reference games, Hawkins et al. (2020a) observed the development of specialized reference phrases for ambiguous shapes, which allows players to reduce their utterances' length and syntactic complexity. Given that CEREALBAR objects are generally unambiguous and familiar, players do not begin with overly verbose references, and have less potential for reduction to more concise references. In contrast, we observe increased instruction length and complexity. Leaders issue an increasing number of tasks to the follower per instruction, utilizing the flexibility afforded by CEREALBAR's design. This less constrained scenario better reflects real-life collaborations, where participants complete many tasks to achieve complex goals.

Our observations show the competing effects of cost-minimization and utility-maximization. The formation of common ground and expectations on partners' behavior enables leaders to use language differently to convey more information-dense instructions to optimize game performance. This is aligned with the expectation of better communication grounding between community members in Clark and Marshall (1981), and with how grounding in Clark and Wilkes-Gibbs (1986) manifests as reduced complexity when utterance utility is fixed. Because there are conflicting forces at work in CEREALBAR, common ground is realized differently.

The most related setup to CEREALBAR is the Cards task (Djalali et al., 2012; Potts, 2012), where two players collect a single set of cards. It uses four static environments and studies dialogue, not instructions. Djalali et al. (2011) showed Cards players increase the interaction complexity by developing a rich common ground, including terms for the fixed board locations. This is less likely with the randomly generated CEREALBAR envi-

ronments. Utterances in Cards also become shorter, potentially due to the predefined number of goals.

Language complexity also increases in communities where users jointly build a natural-language-like programming language (Wang et al., 2017; Gavran et al., 2018). This scenario differs from ours in lacking explicit collaboration on tasks, focusing on a learned programming language rather than natural language, and training a single model, differently from our many-listeners community.

The language dynamics observed in CEREALBAR contrast with those previously observed in reference games, providing evidence that gradual formation of common ground among interaction participants does not necessarily result in reduced complexity of sentences, and may even result in increased complexity. Our conclusions do not void nor mutually exclude previous work, but illustrate the complexity of language change over time in a community. An important direction for future work is controlled studies to observe the effects of scenario design on the interaction between the development of common ground and language change.

Acknowledgments

This research was supported by NSF under grants No. 1750499, 1750499-REU, and DGE-1650441. It also received support from a Google Focused Award, the Break Through Tech summer internship program, and a Facebook PhD Fellowship. We thank Chris Potts and Robert Hawkins for early discussions that initiated this analysis; and Ge Gao and Forrest Davis for their comments.

References

Herbert H. Clark and Catherine R. Marshall. 1981. Definite knowledge and mutual knowledge. *Elements of discourse understanding*, pages 10–63.

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

Alex Djalali, David Clausen, Sven Lauer, Karl Schultz, and Christopher Potts. 2011. Modeling expert effects and common ground using questions under discussion. In *AAAI Fall Symposium: Building Representations of Common Ground with Intelligent Agents*.

Alex Djalali, Sven Lauer, and Christopher Potts. 2012. Corpus evidence for preference-driven interpretation. In *Logic, Language and Meaning*.

Ivan Gavran, Brendon Boldt, Eva Darulova, and Rupak Majumdar. 2018. Precise but natural specification for robot tasks.

Noah D. Goodman and Michael C. Frank. 2016. Pragmatic language interpretation as probabilistic inference. *Trends in Cognitive Sciences*, 20:818–829.

Janosch Haber, Tim Baumgärtner, Ece Takmaz, Lieke Gelderloos, Elia Bruni, and Raquel Fernández. 2019. The PhotoBook dataset: Building common ground through visually-grounded dialogue. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Robert X. D. Hawkins, Michael C. Frank, and Noah D. Goodman. 2020a. Characterizing the dynamics of learning in repeated reference games. *Cognitive science*, 44 6:e12845.

Robert X. D. Hawkins, Mike Frank, and Noah D. Goodman. 2017. Convention-formation in iterated reference games. In *Cognitive Science*.

Robert X. D. Hawkins, Noah D. Goodman, A. Goldberg, and T. Griffiths. 2020b. Generalizing meanings from partners to populations: Hierarchical inference supports convention formation on networks. In *Proceedings of the Annual Conference of the Cognitive Science Society*.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Robert M. Krauss and Sidney Weinheimer. 1964. Changes in reference phrases as a function of frequency of usage in social interaction: a preliminary study. *Psychonomic Science*, 1:113–114.

Will Monroe, Robert X.D. Hawkins, Noah D. Goodman, and Christopher Potts. 2017. Colors in context: A pragmatic neural model for grounded language understanding. *Transactions of the Association for Computational Linguistics*, 5:325–338.

Christopher Potts. 2012. Goal-driven answers in the Cards dialogue corpus. In *Proceedings of the West Coast Conference on Formal Linguistics*, pages 1–20.

Alane Suhr, Claudia Yan, Jack Schluger, Stanley Yu, Hadi Khader, Marwa Mouallem, Iris Zhang, and Yoav Artzi. 2019. Executing instructions in situated collaborative interactions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Takuma Udagawa and Akiko Aizawa. 2019. A natural language corpus of common grounding under continuous and partially-observable context. In *Proceedings of the Conference on Artificial Intelligence*.

Sida I. Wang, Samuel Ginn, Percy Liang, and Christopher D. Manning. 2017. Naturalizing a programming language via interactive learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 929–938, Vancouver, Canada. Association for Computational Linguistics.

Yang Xu and David Reitter. 2016. Convergence of syntactic complexity in conversation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

A Reproducibility Checklist Details

All computation was done on a personal laptop. The CEREALBAR data was acquired from https://github.com/lil-lab/cerealbar.

B Data Details

B.1 Selection of Interactions for Analysis

The data we use was not collected specifically for this analysis, but during data collection for model development by Suhr et al. (2019). We use 795 of the 960 interactions in the original training split of the data for our analysis, pruning the rest to avoid games that include inexperienced players later in the community's life. This prevents the language of novice workers from affecting our analysis after the more experienced community had stabilized, which would potentially suppress convention formation trends observed in existing literature about reference games (Hawkins et al., 2020a). During the original data collection process, after 367 of the 960 total training interactions were collected, the community was split into junior and senior workers. Junior workers became senior upon gaining adequate experience. A junior worker could request to be moved to the senior pool after they had played at least one game as a follower and at least one game as a leader where they earned at least one point with their partner, and they seemed to be following the game rules. Workers who performed well before the split were included in the senior pool. We do not consider games from the junior pool.

B.2 Decile Details

All deciles span a relatively short period of time except the sixth decile, which includes a pause in data collection (Table 2). The pause did not significantly effect community membership or performance. Figure 7 shows the number of instructions per decile, distinguished by complete and incomplete instructions. Incomplete instructions occur at the end of an interaction, when there is insufficient time or turns to complete the instruction. Figure 8 shows mean interaction length in each decile. Figure 9 shows follower path lengths per instruction across each decile.

C Additional Analysis Details

C.1 Interaction Performance

Several measures demonstrate an increase in player expertise. We analyze interaction performance through how many moves are taken per each instruction, the occurrence of de-selection card events, and instruction queuing behavior. We find that followers become better at following instructions and leaders at creating efficient plans.

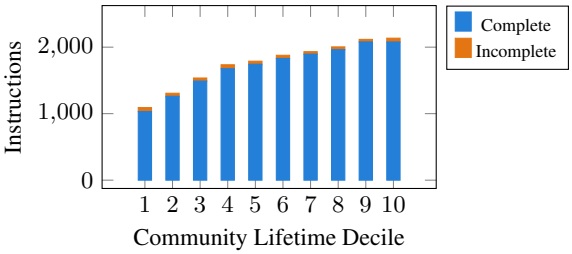

Figure 7: The number of instructions for each decile, distinguished by whether they were marked as complete by the follower.

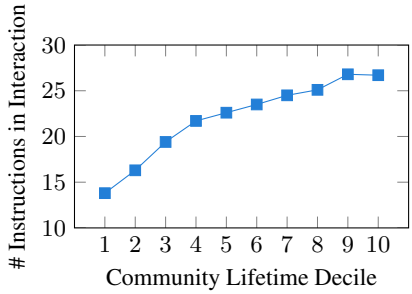

Figure 8: Mean interaction length, measured by the number of instructions, in each decile. We include incomplete instructions in these counts.

Optimal Path Length Deviations We measure how leaders utilize the larger number of steps per turn available to followers through the length of the shortest possible path corresponding to each instruction. We compute this shortest path using the observed start and end positions of the human follower, ensuring that the path avoids obstacles and completes card events completed by the original follower. The mean length of the shortest path per instruction increases over the community lifetime from 6.66 to 7.97 moves ($p < 0.0001$). This corresponds to the increase we observe in the number of goals described in each instruction, which likely requires more steps.

Concurrently, we see improvements in follower instruction execution, measured through the excess moves taken by follower: the difference between the number of moves the follower took and the shortest possible path corresponding to each completed instruction. Over time, the number of excess steps compared to the shortest paths decreased from 3.67 to 2.36 moves ($p < 0.0001$). Figure 10 visualizes this increase in average optimal path length per instruction and decrease in moves taken in ex-

Decile	Game IDs	Lower Time Limit	Upper Time Limit	Time (Days)
1	1-79	2019-01-27 20:05:00 UTC	2019-02-02 15:39:00 UTC	5.815278
2	80-159	2019-02-02 15:39:00 UTC	2019-02-02 20:24:00 UTC	0.197917
3	160-238	2019-02-02 20:24:00 UTC	2019-02-03 00:25:00 UTC	0.167361
4	239-318	2019-02-03 00:25:00 UTC	2019-02-04 00:15:00 UTC	0.993055
5	319-397	2019-02-04 00:15:00 UTC	2019-02-04 03:09:00 UTC	0.120833
6	398-477	2019-02-04 03:09:00 UTC	2019-04-15 19:27:00 UTC	70.6375
7	478-556	2019-04-15 19:27:00 UTC	2019-04-15 23:44:00 UTC	0.178472
8	557-636	2019-04-15 23:44:00 UTC	2019-04-16 20:06:00 UTC	0.848611
9	637-715	2019-04-16 20:06:00 UTC	2019-04-16 22:50:00 UTC	0.113889
10	716-795	2019-04-16 22:50:00 UTC	2019-04-17 03:43:00 UTC	0.203472

Table 2: Time limits of the division into deciles. The last column is the total amount of time elapsed during a decile. All lower time limits are inclusive. All upper time limits are exclusive, except the last one, which is inclusive.

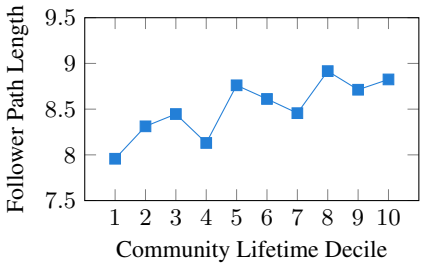

Figure 9: Mean length of observed follower paths for complete instructions in each decile. We measure length in the number of steps recorded per instruction.

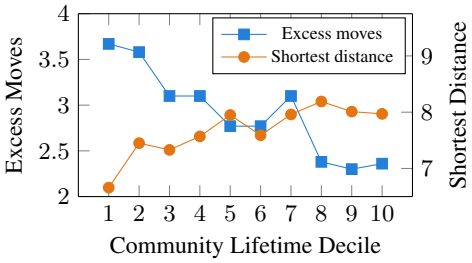

Figure 10: Excess follower moves and shortest possible distance per leader instruction.

cess of this optimal path. The reduction in excess moves is especially notable given the increase in the moves required per instruction, indicating the absolute decrease observed is due to an even higher decrease in the probability of follower errors.

Card De-selections We also study the occurrence of card de-selections, which often reflect error correction. In ideal gameplay, no de-selection events should be observed, as they require additional steps and only correct for a mistakenly selected card not to be part of the current target set. We observe that player errors decrease: the proportion of card events (the selection or de-selection of a single card) that are de-selections decreases from 7.86% to 4.52% ($p = 0.0018$). Figure 11 shows the percentage of card events initiated by either player that are de-selections.

Instruction Queuing The CEREALBAR setup allows a leader to plan ahead by queuing multiple

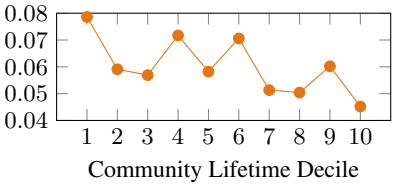

Figure 11: Proportion of all card events, initiated by both followers and leaders, that were de-selections.

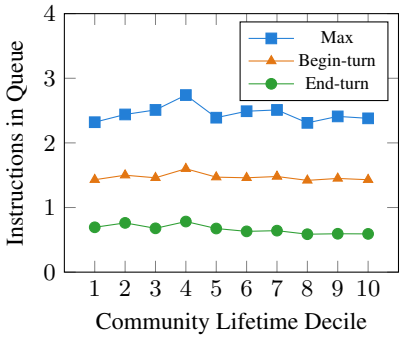

Figure 12: Instruction-queuing behavior over time.

instructions to the follower at a time. For example, to efficiently use all of the follower's moves, a leader may send two instructions: one which tells them to complete the set, and another that tells them to move towards a card which will make up the next set. A larger queue indicates longer-term leader planning. Alternatively, the leader could include the additional information in one instruction without queuing more instructions. We analyze this queuing behavior as a potential alternative explanation: the leaders may improve how they relay information with better planning, rather than changing the content of their instructions.

We measure the size of the queue at the beginning and end of follower turns, and the maximum queue size reached during a game. Figure 12 shows queue statistics over time. Begin-turn queue size directly measures how leaders plan via queuing instructions, as no instructions are queued during the follower's turn. Begin-turn and maximum queue size did not change significantly over

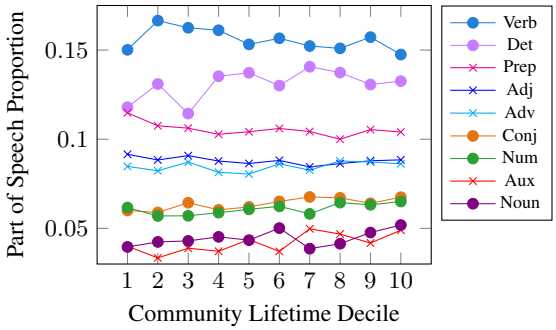

Figure 13: Ratio of language that is a specified part of speech over time. Parts of speech of particular interest are plotted with filled markers.

Dep = 0.83, Wid = 0.93, Bch = 0.83
turn to the left to see one yellow sqaure

Dep = 1.14, Wid = 1.03, Bch = 0.96
go forward one and to your left is orange

Dep = 1.58, Wid = 0.66, Bch = 0.65
take the green card with 3 symbols in front of you

Dep = 0.79, Wid = 1.26, Bch = 1.01
Head straight towards the blue plus card, but don't pick it up. Continue past it, on the left of it.

Figure 14: Selected instructions to illustrate the different measures of complexity, namely: maximum depth (dep), maximum width (wid), and average branching factor (bch). All measures normalized for length.

time. This relative stability indicates that game play improvements were not primarily due to leaders planning ahead across separate instructions; rather, they can be attributed more to the changes of language within instructions. End-turn queue size sampling indicates the efficiency of player collaboration. From the first to last decile, the average end-turn queue size decreases from 0.694 to 0.592 instructions. This indicates that followers become more efficient over time, completing more instructions per turn. This aligns with our analysis of follower efficiency (Section C.1 and Figure 10).

C.2 Syntactic Complexity

Part-of-Speech Analysis To compute the ratio of POS use, we treat each decile of community life as a bag of words, dividing the total tag count of each POS by the total token count in each decile. In our analysis, we combine the spaCy tags ⟨sconj⟩ (subordinating conjunction) and ⟨cconj⟩ (coordinating conjunction) into one conjunction class, and the tags nouns and proper nouns into one noun class. Figure 13 shows the proportion of the nine most common POS tags used in CEREALBAR instructions: verbs, determiners, prepositions, adjectives, adverbs, conjunctions, numerals, auxiliary verbs, and nouns.

Class	Keywords
Road	*road*, *fork*, *path*, *intersect*, *trail*, *crossroad*, *crosspath*, *walkway*
Foliage	*palm*, *flower*, *tree*, *shrub*, *grass*, *pine*, *bush*, *grove*, *plant*, *conif*, *field*, *foliag*, *wasteland*, *forest*, *clearing*, *patch*, *lawn*
Building	*tower*, *building*, *house*, *tent*, *barn*, *fort*, *doghouse*, *hut*, *village*, *cabin*, *shack*, *structure*, *shed*, *tower*
Water	*lake*, *pond*, *water*, *sea*, *river*, *coast*, *island*, *shore*
Rock	*rock*, *cliff*, *boulder*, *mountain*, *hill*, *log*, *stone*
Ice	*glacier*, *ice*, *iceberg*
Light	*post*, *lamp*, *pole*, *light*

Table 3: Reference class keywords

Syntactic Complexity Analysis For each utterance, we measure the branching factor, maximum width, and maximum depth of its dependency parse. Dependency tree depth indicates how many embedded clauses the utterance has, whereas width-related measures indicate how many modifiers are stacked in one sub-tree. Intuitively, increased width-related metrics indicate more descriptive utterances, whereas increased depth indicates more compounded phrases. Figure 14 provide examples to illustrate these differences.

We normalize these measures by the utterance length following Xu and Reitter (2016). Formally, let X_n be the set of all utterances in our data with a length of n tokens. The average of metric S (e.g., maximum width) across all utterances of length n in our data is:

$$\overline{S}(n) = \frac{1}{|X_n|} \sum_{x \in X_n} s(x) \quad . \tag{1}$$

For each utterance x with length n, we compute the normalized measure for the utterance:

$$s'(x) = \frac{s(x)}{\overline{S}(n)} \quad . \tag{2}$$

C.3 Reference Change

We divide environmental objects in the CerealBar game into six classes: road, foliage, building, water, rock, ice, and light class objects. We use regular expressions to automate if an utterance refers to a class of objects, defined by if it contains at least one of the class keywords in Table 3.