Studying Alignment in Spontaneous Speech via Automatic Methods: How Do Children Use Task-specific Referents to Succeed in a Collaborative Learning Activity?

Utku Norman∗, UTKU.NORMAN@EPFL.CH
CHILI Lab, EPFL, Lausanne, Switzerland

Tanvi Dinkar∗, TANVI.DINKAR@TELECOM-PARIS.FR
IDS, S2A Lab, Télécom Paris, Paris, France

Barbara Bruno, BARBARA.BRUNO@EPFL.CH
CHILI Lab, EPFL, Lausanne, Switzerland

Chloé Clavel, CHLOE.CLAVEL@TELECOM-PARIS.FR
IDS, S2A Lab, Télécom Paris, Paris, France

Abstract

A dialogue is successful when there is alignment between the speakers, at different linguistic levels. In this work, we consider the dialogue occurring between interlocutors engaged in a collaborative learning task, and explore how performance and learning (i.e. task success) relate to dialogue alignment processes. The main contribution of this work is to propose new measures to automatically study alignment, to consider completely spontaneous spoken dialogues among children in the context of a collaborative learning activity. Our measures of alignment consider the children’s use of expressions that are related to the task at hand, their follow up actions of these expressions, and how it links to task success. Focusing on expressions related to the task gives us insight into the way children use (potentially unfamiliar) terminology related to the task. A first finding of this work is the discovery that the measures we propose can capture elements of lexical alignment in such a context. Through these measures, we find that teams with bad performance often aligned too late in the dialogue to achieve task success, and that they were late to follow up each other’s instructions with actions. We also found that while interlocutors do not exhibit hesitation phenomena (which we measure by looking at fillers) in introducing expressions pertaining to the task, they do exhibit hesitation before accepting the expression, in the role of clarification. Lastly, we show that information management markers (measured by the discourse marker ‘oh’) occur in the general vicinity of the follow up actions from (automatically) inferred instructions. However, good performers tend to have this marker closer to these actions. Our measures still reflect some fine-grained aspects of learning in the dialogue, even if we cannot conclude that overall they are linked to the final measure of learning. Our measures contribute to the research in the field of situated dialogue and towards bridging the gap between dialogue and collaborative learning research.

Keywords: alignment; situated dialogue; collaborative learning; spontaneous speech; disfluency

∗The authors contributed equally to this research.

1The anonymised “dataset” (JUSThink Dialogue and Actions Corpus), as well as the “tools” for our analyses in this paper (JUSThink Alignment Analysis), are publicly available online, from the Zenodo Repositories DOI: 10.5281/zenodo.4627104 for the dataset, and DOI: 10.5281/zenodo.4675070 for the tools.
1. Introduction

Collaboration occurs in a situation in which individuals work together as members of a group in order to solve a problem (Roschelle and Teasley [1995]). In a collaborative activity, members build shared, abstract representations of the problem at hand (Schwartz [1995]). Collaborative activities have been analysed by focusing on the verbal (spoken communication) or the non-verbal (e.g. gaze, facial expressions, gestures) behaviours of the group members. Research has found that non-verbal cues could indicate the quality of collaboration, for example the link between high coupling of a pairs’ focus of attention and high quality of collaboration (Jermann et al. [2011]), the link between automatically annotated eye gaze coupling and interaction quality (Jermann and Nüssli [2012]), and the link between non-verbal cues like mutual gaze and laughter, and group cohesion (Bangalore Kantharaju et al. [2020]). In terms of verbal behaviour, it was found that by automatically labelling silent and spoken episodes, the amount of speech is higher for pairs who had a higher quality of interaction (Jermann and Nüssli [2012]). Since a spoken episode could contain any kind of speech, this study was limited, and the authors conclude the need to analyse verbal behaviours in greater detail in the study of collaborative activities.

When a collaborative activity is in an educational setting, it is termed as collaborative learning: where individuals ‘learn’ together. The group interactions in collaborative learning are expected to activate mechanisms that would bring about learning, although there is no guarantee that these beneficial interactions will happen (Dillenbourg [1999]). Collaborative learning involves group processes that are carried out interactively and hence it contains, but it is irreducible to, individual learning (Stahl et al. [2006]). There is a need for new tools to analyse these interactions by treating them as group processes, to better understand different learning mechanisms (Dillenbourg et al. [1996]).

The verbal analysis of collaborative learning tends to be based on manual coding schemes that distinguish between various types of verbal interactions. For example, Visschers-Pleijers et al. (2006) labelled several types of verbal interactions, such as asking exploratory questions and handling conflicts about knowledge, and inquired into how they are distributed over time in a collaborative learning activity. Similarly, Yew and Schmidt (2009) used a more extensive coding scheme to qualitatively study the nature of verbal interactions. Yew and Schmidt (2012) also manually labelled and counted the number of relevant concepts that were verbalised, and related this to learning outcomes. However, we need automatic tools that analyse verbal interactions in a collaborative learning activity as, so far, qualitative analyses require extensive manual annotation.

In parallel, there have been studies that do not focus on educational goals and collaborative learning, but on how humans understand each other by analysing human-human dialogues. Similar to understanding group processes in collaborative learning, studying a dialogue is challenging; as rather than an individual effort, a dialogue is an interactive activity performed by two or more people, i.e. interlocutors (Clark and Wilkes-Gibbs [1986]). Interlocutors ideally take turns to try to reach a common or mutual understanding (Clark and Schaefer [1989]). Dissimilar to collaborative learning, works that study mutual understanding largely consider dialogues amongst adult interlocutors solving a problem together e.g. in (Garrod and Anderson [1987], Fusaro and Tylén [2016]), without the added dimension of a learning goal. However, task performance is not always positively correlated with learning outcomes, as a student can fail in the task but learn from it (even learn from failure, as in “productive failure” (Kapur [2008], Kapur and Bielaczyc [2012])) and perform well in the task but not learn from it (i.e. unproductive performers (Kuhn [2015])). Thus, works on collabor-
orative learning extensively study behaviours related to learning outcomes, but can lack in-depth dialogue analysis, and automatic tools are missing. On the other hand, several works on dialogue, particularly on mutual understanding, have extensively studied the dialogue, but without the added depth of learning outcomes. Thus, we need to adapt both the activity as well as the tools of analysis in order to accommodate the learning objectives.

The objective of this article is to bring together the two perspectives, to contribute to the exploration of the deep, complex and tangled relationship between what we say and what we do, and the outcomes of this. Specifically, we are interested in the form this triangle takes when applied to children engaged in a collaborative learning activity, in which what they say and do is strongly tied to how they perform, and subsequently what they will ultimately learn from the activity. To investigate the above question, we transcribe a subset of previously collected data (containing audio and action logs) generated by teams of two children engaged in a collaborative learning activity. This activity aimed at providing an intuitive understanding of networks/graphs and spanning trees, in order to improve the Computational Thinking (CT) skills of children (Nasir et al. 2020c; Nasir, Norman, et al., 2020b). This activity was deliberately designed so that to further the task, the children needed to understand each other. This corpus is called the JUSThink Dialogue and Actions Corpus.

From a methodological perspective, the automatic analysis of verbal behaviour in a collaborative learning activity and its connection to task success (i.e. task performance and learning outcomes, collectively) is extremely challenging for two reasons. Firstly, existing methods for studying collaboration rely on time-consuming manual annotations of conversations. Here, we want to provide automatic tools for studying collaborative processes. Second, we tackle the problem of the analysis of spontaneous conversational speech of children, which has its own features, which could be more difficult to handle and less-structured than adults’ speech. We propose automatic methods to investigate how children refer to their environment, instruct each other, and take actions together while solving the task. Specifically we consider the children’s i) use of expressions related to the task, and the ii) follow-up actions of such expressions. We then ask how this links with their task success, i.e. task performance and learning outcomes. Our work highlights the need to bridge the gap between the fields of collaborative learning and dialogue, and expand existing automatic measures for verbal behaviours, as they can have an impact on our understanding about the way children learn. The following section illustrates the kinds of verbal behaviours we expect to observe in the collaborative learning activity based on dialogue literature.

Background

Referring in a situated environment. In a collaborative activity, interlocutors often work together to achieve success in a task. When these activities are situated, the verbal communication has an additional dependence on the immediate environment in which the dialogue takes place (Kruijff et al., 2010). This entails interlocutors referring to the physical environment. Often, referring in such dialogues manifests in action, where in order to take an appropriate action, interlocutors must understand (to some degree) the other’s way of referring to their environment (Clark and Brennan, 1991). Consider the following example of a situated dialogue, taken from the JUSThink Dialogue and Actions Corpus. Two interlocutors had to work together to construct a solution in a situated

---

2We use the term dialogue literature to refer to relevant linguistic (e.g. psycholinguistics, discourse, conversation analysis) literature.
activity, by deciding to connect (or disconnect) mountains on a given map together.

A: What about Mount Davos to Mount, Saint Gallen?
B: Because what if, you did if we could do it?
A: What about Mount um Davos to Mount Gallen?
B: Mount . . .
B: oh Mount Davos
A: yeah to Mount Gallen.
B: to Mount Gallen yeah do that.
〈A connects Mount Gallen to Mount Davos〉
B: Okay my turn.

A’s way of referring to a landmark in the environment (e.g. Davos – “Mount Davos”) was understood by B as the same landmark (Davos – “oh Mount Davos”), as evidenced by repeating it. Thus, interlocutors form referring expressions, like “Mount Davos”, which are pointers to objects, which we call task-specific referents (here, the landmark Davos), that have been referred to within the situated environment. Thus, the materialisation from referring (saying) to action (doing) is immediately apparent in such dialogues, and affects task success. In the above Example 1 because A and B mutually understood the other’s use of referring expressions; they make progress in drawing the route correctly.

Medium of communication and spontaneous speech. Situated dialogues can happen through constrained mediums of communication, for example chatting via a text environment e.g. in (Stahl 2007; Dillenbourg and Traum 2006). We focus on face-to-face communication, due to the co-presence of the interlocutors, visibility and audibility in the medium, which can dramatically affect the process of mutual understanding (Dillenbourg and Traum 2006; Clark and Brennan 1991). Since we focus on face-to-face communication, spontaneous speech phenomena becomes a relevant dimension of analysis in verbal behaviour.

In the study of referring expressions in a situated face-to-face environment, we aim to account for the referring expression and also the surrounding spontaneous speech phenomena. The aim is to study both what the interlocutors said in essence and how it was said. In spoken language understanding, collapsing an input utterance into its lexical level is standard practice (for example, as seen in (Tur and De Mori 2011, Ch. 13: Speech Summarisation)). However, these expressions do not occur in a vacuum, they are part of a spoken dialogue amongst interlocutors, which is ‘messy’ (disfluent) especially when considering children’s speech.

Thus the analysis of spontaneous speech in this context is often neglected, but could be illuminating on a conversational level. This is pertinent when considering the added dimension of a pedagogical goal, as certain speech phenomena have been linked to signs of hesitation and uncertainty (Pickett 2018; Smith and Clark 1993; Brennan and Williams 1995) and information management (Schiffrin 1987). We believe that by just narrowing on these, we will get further insight into the way children refer to their situated environment and thus learn.

Grounding and alignment. Mutual understanding, at least to some degree can be attributed to a process called grounding (Clark and Schaefer 1989; Clark and Brennan 1991), which can be thought of as adding to what is already mutually understood. We have currently defined this notion imagining that interlocutors contribute to their common ground, i.e. build up a shared mutual understanding. However, Pickering and Garrod (2004) stress on the importance of an automatic priming
mechanism, i.e. when a listener encounters an utterance from the speaker, it is more likely that subsequently the listener will produce an utterance by using this representation. Here, the listener need not explicitly assent-to/accept the speaker’s contribution; instead, there can simply be an implicit common ground unless a misunderstanding requires changing the representation. Like this, routines, or referring expressions that are “fixed”, become shared or established amongst interlocutors. Thus rather than grounding, they define alignment of representations, to mean the “development of similar representations in the interlocutors” [Pickering and Garrod, 2006]. Therefore the interlocutors succeed in understanding each other when there is alignment between them, or a shared representation, at different linguistic levels. Pickering and Garrod [2004, 2006] in a maze task found that at first glance, the dialogue seemed unstructured, but then found underlying alignment structure that supported collaboration amongst the interlocutors.

Dubuisson Duplessis et al. [2017, 2021] proposed automatic and generic measures to extract lexical structures of alignment (which they refer to as verbal alignment) in a task-oriented dialogue. The proposed method works on alignment based on surface matching of text and does not focus on other levels of linguistic alignment as envisioned in [Pickering and Garrod, 2004]. However, it is done with the aim of automatically finding these text patterns in the dialogue, by sequentially processing a transcript in an unsupervised manner. In this article, we provide a new tool/framework for studying situated dialogue building on the automatic and generic methodology by Dubuisson Duplessis et al. [2017, 2021]: this implies to model how the interlocutors refer to their environment and to extend the tool based on this model. We then propose new measures that allow us to study another level of alignment in situated dialogues; i.e. via automatically inferring instructions given by the interlocutors, and then linking those instructions to actions taken.

**Aims and research questions.** Following these lines of inquiry, in this article we investigate whether alignment between children in dialogues that emerge from a collaborative learning activity is linked with success in the task.

Concretely, we investigate the following Research Questions:

- **RQ1:** How do the interlocutors use task-specific referents? Does this link to task success?

- **RQ2:** How do the interlocutors follow up the use of task-specific referents with actions? Does this link to task success?

Thus, we study specifically the task-specific referents required to perform in the activity and its link to task success (performance and learning scores). Though our results are on a small dataset of 10 transcribed dialogues, the proposed method automatically captures some patterns of alignment on completely spontaneous children’s dialogues, and its relation to task success, without time-consuming manual annotations. The rest of the article is organised as follows. Sec. 2 describes our context: the activity and the dataset we obtained to study our research questions. Sec. 3 presents our methodology with our hypotheses and how we investigate them. Sec. 4 presents our results and our interpretations that address RQ1 and RQ2. Sec. 5 concludes the findings of the paper.
2. Materials

2.1 Collaborative problem solving activity, scenario and setup

JUSTThink is a collaborative problem solving activity for school children (Nasir et al., 2020c; Nasir, Norman, et al., 2020b). It aims to improve their CT skills by exercising their abstract reasoning on graphs. Recent research on educational curricula stresses the need for learning CT skills in schools, as going beyond simple digital literacy to developing these CT skills becomes crucial (Menon et al., 2019). With this in mind, the objective of the activity is to expose school children to minimum-spanning-tree problems.

Scenario. A humanoid robot, acting as the CEO of a gold mining company, presents the activity to the children as a game, asking them to help it collect gold, by connecting gold mines one another with railway tracks. They are told to spend as little money as possible to build the tracks, which change in cost according to how they connect the gold mines. The goal of the activity is to find a solution that minimises the overall cost, i.e. an optimal solution for the given network.

Children participate in teams of two to collaboratively construct a solution, by drawing and erasing tracks. Once all gold mines are reachable, i.e. in some way connected to each other, they can submit their solution to the robot for evaluation. They must submit their solution together, and can submit as many times as they want. The robot then reveals whether their solution is an optimal solution or, if not, how far it is from an optimal solution (in terms of its cost). In the latter case, children are also encouraged by the robot to try again. They can submit a solution as many times as they want until the allotted time for the activity is over.

Setup. Two children sit across each other, separated by a barrier. A touch screen is placed horizontally in front of each child. Children can see each other, but cannot see the other’s screen, as shown in Fig. 1. They are encouraged by the robot to verbally interact with each other, and work together to construct a solution to the activity.

The screens display two different views of the current solution to the children. One view is an abstract view, where the gold mines are represented as nodes, and the railway tracks that connect them as edges (see Fig. 2a). The other view, or the visual view, represents the gold mines and railway tracks with images (see Fig. 2b). A child in the abstract view can see the cost of built edges, but cannot act upon the network. Conversely, in the visual view, a child can add or delete an edge, which is a railway track, but cannot see its cost. The views of the children are swapped every two edit actions, which is any addition or deletion of an edge. Hence, after every two edit actions, the child that was in the abstract view moves to the visual view and vice versa. A turn is thus the time

3The minimum-spanning-tree problem is defined on a graph $G = (V, E)$ with a cost function on its edges $E$, where $V$ is the set of nodes, $E \subseteq V \times V$ is the set of edges that connects pairs of nodes. The goal is to find a subset of edges $T \subseteq E$ such that the total cost of the edges in $T$ is minimised. This corresponds to finding a minimum-cost sub-network of the given network.


5Note that we treat this triadic activity as a dyadic dialogue. The children are initially prompted by the robot to work with each other, and later simply given the cost of their sub-optimal solution. However, almost all of the exchanges are between the two interlocutors. After careful observation of the dialogues in the dataset, we observe the tendency to ignore the robot unless submitting a solution.
STUDYING ALIGNMENT IN SPONTANEOUS SPEECH VIA AUTOMATIC METHODS

Figure 1: The JUSTThink activity setup.

![Figure 1: The JUSTThink activity setup.](image)

(a) An example abstract view. (b) Corresponding visual view.

Figure 2: interlocutors’ views during the JUSTThink activity

interval between two view swaps, i.e. in which one child is in the visual view and the other child is in the abstract view. A turn lasts for two edit actions. This design aims at encouraging children (interlocutors) to collaborate.

This activity is particularly suited to study alignment, since it is designed in such a way to create interdependence, i.e. a mutual reliance to further the task, between the interlocutors. This interdependence requires interlocutors to align with each other on multiple levels, e.g. how to refer to the environment and how to represent the activity, in order to succeed. Concretely, the activity enables:

- **Swapping and visual view control.** Since a turn changes every two edit actions, if an interlocutor has a particular action they want to take, they have to either wait for their turn in the visual view to implement the desired change, or instruct the other interlocutor. Here, we follow an idealised perspective on the activity: at a given time, the interlocutor in the abstract view is an Instruction Giver (IG) who describes their instructions for the task by using specific referring expressions, and the other (in visual view) is the Instruction Follower (IF) who
executes the action (this is akin to the Map Task\cite{Anderson1991}). The activity design creates a frequent swapping of views. This aims to discourage interlocutors from working in isolation or in fixed roles of IG and IF, which could potentially happen in collaborative tasks.

- **Routine expressions and alignment in the task.** Since the robot uses brief and general instructions to present the activity and its goal, the interlocutors must figure out for themselves the way to approach the activity. The task-specific referents, which are the names of the gold mines (cities in Switzerland, a multi-lingual country\footnote{The names of the gold mines are potentially unfamiliar to interlocutors. Interlocutors must refer to the task, and then align with the other to form routine expressions. By aligning, they establish a shared lexicon, and thus align their representations of the activity with each other.}), are potentially unfamiliar to interlocutors. Interlocutors must refer to the task, and then align with the other to form routine expressions. By aligning, they establish a shared lexicon, and thus align their representations of the activity with each other.

- **Submission of solutions.** Since the interlocutors have to submit their solution together by pressing the “submit” button that is present in both views, they have to, at least, align in terms of their intent to submit. Alternatively, one interlocutor must convince the other to reach a common intent.

### 2.2 The Dataset

**Dataset description.** We collected the JUSTThink dataset\footnote{The anonymised version of the dataset (JUSTThink Dialogue and Actions Corpus), that includes dialogue transcripts, event logs, and test responses, is publicly available online, from the Zenodo Repository DOI:10.5281/zenodo.4627104.}, which consists of 76 children in teams of two (41 females: \(M = 10.3, SD = 0.75\) years old; and 35 males: \(M = 10.4, SD = 0.61\)). We have one problem solving session, i.e. task, per team. 8 out of the 38 teams (\(\approx 21\%\)) found an optimal solution to the activity. The teams were formed randomly, without considering the gender, nationality, or the mother tongue. They include mixed and same gender pairs, and this information is available but not used in our analyses. The study was conducted in multiple international schools in Switzerland, where the medium of education is in English, and hence students are proficient in English. The dataset contains:

1. **The recorded audio files:** Audio was recorded as two mono audio channels synchronised to each other, with one lavalier microphone per channel. The interlocutors were asked to speak in English. The microphones were clipped onto the interlocutors’ shirts.

2. **Event log files:** Event log entries consist of timestamped touch and button press events, application status changes, and the interlocutor’s edits to the tracks.

3. **Pre-test and post-test:** Interlocutors’ responses to the items in the pre-test and post-test\footnote{See the App. B for more details on the pre-test and the post-test.}

To measure performance in the task, we use the cost of the solution submitted at each attempt, extracted from the event log files. To compute a measure of learning outcomes, we use interlocutors’ scores in the pre-test and the post-test (see Sec. \[3.2\] for the measure we adopt). To study how are task-specific referents are used (RQ1), we focus on the recorded audio files and transcribe a representative subset. For (RQ2), to see how do the interlocutors follow up the use of task-specific referents with actions, we combine the edit actions from the log files the with the transcripts.

\footnote{In fact, the IF could also be following their own intuitions and ignoring the IG. Yet, ultimately we think this is a justified assumption, as only the IF is in control of the actions.}
Figure 3: Scatter plot of the transcribed teams (red dots) and non-transcribed teams (blue dots) in the learning outcome (learn) vs. task performance (error) space. The mean of a set (transcribed or other) is shown as a dashed line. Lines indicate the fit of a univariate kernel density estimate for the corresponding set. Numbers denote the ID of teams.
Table 1: Descriptive statistics for the transcribed teams ($N = 10$). SD stands for standard deviation.

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>SD</th>
<th>min</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of submitted solutions</td>
<td>9.4</td>
<td>4.7</td>
<td>4</td>
<td>19</td>
</tr>
<tr>
<td>number of turns in task</td>
<td>50.4</td>
<td>21.4</td>
<td>28</td>
<td>98</td>
</tr>
<tr>
<td>total duration (mins)</td>
<td>23.7</td>
<td>7.1</td>
<td>11.2</td>
<td>36.0</td>
</tr>
<tr>
<td>time per submission (mins)</td>
<td>2.6</td>
<td>2.3</td>
<td>0.7</td>
<td>12.8</td>
</tr>
<tr>
<td>duration of a turn (secs)</td>
<td>25.0</td>
<td>30.3</td>
<td>1.1</td>
<td>240.2</td>
</tr>
<tr>
<td>length of utterance (in tokens)</td>
<td>6.6</td>
<td>5.4</td>
<td>1</td>
<td>32</td>
</tr>
</tbody>
</table>

**Transcribing a representative subset.** In order to study referring and alignment of the interlocutors, we selected a *subset* of 10 teams of the dataset. The teams were chosen to be a representative sample of the overall dataset (see Fig. 3), keeping in mind the task success distribution; which we measure through performance in the task and learning outcomes observed from the pre-test and the post-test. Table 1 provides further details about the transcribed subset. As shown, the mean duration of the task is $\approx 23$ minutes, and the transcriptions account for $\approx 4$ hours of data.

The transcripts report which interlocutor is speaking (either A or B) and the start and end timestamps for each utterance, beside the utterance content. Utterance segmentation is based on Koiso et al. (1998)’s definition of an *Inter Pausal Unit (IPU)*, defined as “a stretch of a single interlocutor’s speech bounded by pauses longer than 100 ms”. We also annotated punctuation markers, such as commas, full stops, exclamation points and question marks. Fillers, such as ‘uh’ and ‘um’, (using Meteer et al. (1995) for reference) were transcribed, as well as the discourse marker ‘oh’. The above 3 spontaneous speech phenomena occur frequently in the dataset { ‘um’: 236, ‘uh’: 173, ‘oh’: 333}. Other phenomena, such as ‘ew’ or ‘oops!’ were also transcribed, however, their frequency is too low for analysis. Transcription included incomplete elements, such as “Mount Neuchat- um Mount Interlaken”.

Pronunciation differed among and within interlocutors (for example, for the word ‘Montreux’, pronouncing the ending as /ks/ or /ø/), due to the unfamiliarity of the interlocutors with the referents, and individual accents. As our methodology is dependent on matching surface forms (refer to Sec. 3), we did not transcribe pronunciation variants of a word. A graduate student completed two passes on each transcript, which were then checked by another native English speaker graduate student with experience in transcription/annotation tasks.

---

9 We relied on manual transcription, due to the poor performance of state-of-the-art automatic speech recognition systems on this dataset which consists of children’s speech with music playing in the background. The transcripts are publicly available online (in the transcripts/ folder, available from the Zenodo Repository, DOI: 10.5281/zenodo.4627104)

10 See App. A for details on the selection.

11 ‘Oh’ can also be treated as a filler in Meteer et al. (1995), but we consider the usage of ‘oh’ more aligned with Schiffrin (1987).

12 We standardise variations of pronunciation in the transcriptions, and we do not account for e.g. variations in accent.
3. Methodology

3.1 Structure of the Paper and the Hypotheses

The present work focuses on the task-specific referents that interlocutors minimally require to succeed in the task. Therefore, we restrict the possible referring expressions to ones that contain task-specific referents, in particular, to the objects that the interlocutors are explicitly given on the map (see Fig. 2). Interlocutors only need this terminology with certain function words to progress in the task (e.g. “Montreux to Basel”). We believe that this design choice is particularly suited to study alignment in this type of activity, as the frequent swapping of views encourages the interlocutors to communicate with the other their intents using these referents. This allows us to focus on verbal contributions that are explicitly linked to the ‘situatedness’ of the task and it’s relation to a final measure of task success. While there are certainly other referring expressions to consider (e.g. “That mountain there”) not containing task-specific referents, it would require some degree of manual annotation. We thus focus on task/domain specific referents that can be automatically extracted.

With this in mind, RQ1 considers the use of task-specific referents, while RQ2 considers the follow-up actions taken after these task-specific referents were uttered. Thus RQ1 focuses solely on the surface-level utterances, i.e. “What did the interlocutors say”, while RQ2 builds on this with actions, i.e. “What did the interlocutors do afterwards”. Since the nature of the activity is situated, we look into whether the interlocutors are stuck saying task specific referents, or whether the use of these referents spurs the interlocutors into action; i.e. into doing. We investigate RQ1 and RQ2’s relation to task success, which is evaluated by two measures: i) performance, and ii) learning outcomes. Please refer to Fig. 4 for an overview of the RQs.

RQ1: How do the interlocutors use task-specific referents? Does this link to task success? In RQ1, we specifically consider the link between the use of task-specific referents and task success through the routines’ i) temporality, and ii) surrounding hesitation phenomena. We expand on these in Sec. 3.3. Specifically, we hypothesise:

- **H1.1: Task-specific referents become routine early for more successful teams.** We expect better performing teams to establish routine expressions earlier in the dialogue. Ideally by quicker establishment of routine expressions, teams will understand each other faster and thus perform better in the task, and learn more.

- **H1.2: Hesitation phenomena are more likely to occur in the vicinity of priming and establishment of task-specific referents for more successful teams.** We expect that new contributions to the dialogue, via priming (the speaker first introduces the referent) and establishment (the listener utilises this referent for the first time) of routine expressions are associated with hesitation phenomena. A prolonged occurrence of hesitation phenomena not associated with the priming or establishment of routine expressions could highlight greater lack of understanding of the task, and hence be related to lower task success.

RQ2: How do the interlocutors follow up the use of task-specific referents with actions? Does this link to task success? RQ2 investigates how the use of task-specific referents manifests in the interlocutors’ actions within the task, and whether this affects their task success. For this purpose, we consider the instructions of an interlocutor, as the verbalised instructions one interlocutor gives to the other, which we extract through their use of task-specific referents. A physical manifestation
of this instruction could result in a corresponding edit action\textsuperscript{13} or a different edit action. We investigate the follow-up actions of the task-specific referents with its effect on task success, through the follow-up actions’ i) temporality, and the ii) surrounding information management phenomena. We expand on these in Sec. 3.4. We hypothesise:

• **H2.1:** Instructions are more likely to be followed by a corresponding action early in the dialogue for more successful teams. We expect that the earlier interlocutors align with each other in terms of instructions and follow-up actions, the better they progress in the task, and the greater the chance of success in the task. This idea of verbalised instructions being followed up by corresponding actions is in line with previous research on alignment (i.e., interlocutors being in alignment in a successful dialogue, see Sec. 1). However, work on collaborative learning suggests that individual cognitive development (in our case, positive learning outcomes) happens via socio-cognitive conflict (Mugny and Doise, 1978; Doise and Mugny, 1984), and its regulation (Butera et al., 2019). In our task, this means a verbalised instruction could be followed by a corresponding or a different action; as a different action could result in collaboratively resolving conflicts and together building a solution – resulting in task success. It is thus the required effort to construct the shared understanding together that results in collaborative learning (Dillenbourg et al., 2009). For alignment purposes we focus on corresponding actions, but also follow-ups with a different action to gain a clearer picture into the alignment processes of the interlocutors.

• **H2.2:** When instructions are followed by a corresponding or a different action, the action is more likely to be in the vicinity of information management phenomena for more successful teams. The activity involves the creation of a joint focus of attention and the management of information states that are likely to be accompanied by information management phenomena, such as the use of ‘oh’ as an information marker (Schiffrin, 1987). Ultimately, an increase in shared information should lead to an increase in task success.

\textsuperscript{13}Note that there need not be an instruction that precedes an action, i.e. an interlocutor in the visual view can take less-informed actions by him/herself, see Sec. 2.1.
3.2 Measuring Success in the Task

**Measuring task performance.** We adopt a measure that is based on the costs of the submitted solutions compared to the cost of an optimal solution (which is always the same). For each proposed solution, we calculate the normalised cost as the difference between the cost of the solution and the cost of an optimal solution, normalised by the cost of the optimal solution. We define $\text{error}$ as the smallest normalised cost value among all submitted solutions, which represents the team’s closest solution to an optimal solution. We use $\text{error}$ to measure task performance.\(^{14}\)

**Measuring learning outcomes.** Learning measures commonly build upon the difference between the post-test and pre-test results, e.g. in Sangin et al. (2011); which indicates how much an interlocutor’s knowledge on the subject has changed due to the activity. We measure the learning outcomes on the basis of the relative learning gain ($\text{learn}_P$) of an interlocutor $P$, which essentially is the difference between pre-test and post-test, normalised by the margin of improvement or decline $\text{learn}_P$. It is computed as:

\[
\text{learn}_P = \begin{cases} 
\frac{\text{post} - \text{pre}}{10 - \text{pre}} & \text{post} \geq \text{pre} \\
\frac{\text{post} - \text{pre}}{\text{pre}} & \text{post} < \text{pre}
\end{cases}
\]

(2)

where 10 is the maximum score available. It indicates how much the interlocutor learnt as a fraction of how much the interlocutor could have learnt. We use $\text{learn}_P$, the average relative learning gain of both interlocutors, to measure a team’s learning outcomes.\(^{15}\)

3.3 Studying the Interlocutors’ Use of Task-specific Referents (RQ1)

**Detecting routines of task-specific referents.** We formally define a routine expression (adapted from Dubuisson Duplessis et al. (2017, 2021); Pickering and Garrod (2004)) as a referring expression shared by two interlocutors if i) the referring expression is produced by both interlocutors, and ii) it is produced at least once without being part of a larger routine. A routine is based on the reuse of a referring expression, but is specific to the exact matching of token sequences in two utterance strings. Using the above terminology, ‘Montreux’ is a task-specific referent, and an interlocutor might prime the referring expression “Mount Montreux”. If the other interlocutor reuses this referring expression, it thus becomes routine. In particular, we define the utterance at which a referring expression becomes routine as the establishment of that routine. We extract routines and the utterances at which the routines are primed and established from the transcripts as in (Dubuisson Duplessis et al., 2017, 2021). Then, we filter for the routines that contain a task-specific referent.\(^{16}\)

\(^{14}\)We process logs (folder logs/ in the dataset at DOI: 10.5281/zenodo.4627104) with a script (tools/extract_performance_and_other_features_from_logs.ipynb in the tools at DOI: 10.5281/zenodo.4675070) to compute the $\text{error}$ for each team (in table processed_data/log_features/justthink19_log_features_task_level.csv, available with the tools).

\(^{15}\)We process test responses (folder test_responses/ in the dataset) with a script (tools/extract_learning_gain_from_test_responses.ipynb in the tools) to compute the $\text{learn}_P$ for each team (in table processed_data/learning_features/justthink19_learning_features.csv, available with the tools).

\(^{16}\)We process the transcripts (folder transcripts/ in the dataset at DOI: 10.5281/zenodo.4627104) with a script (tools/extract_routines_from_transcripts.ipynb in the tools at DOI: 10.5281/zenodo.4675070) to extract routines (in tables in folder processed_data/routines/, available with the tools). We use the dialign package v1.0, available at https://github.com/GuillaumeDD/dialign by Dubuisson Duplessis et al. (2017, 2021).
Studying the temporality of the establishment of the routines (H1.1). To investigate when the routine expressions become established, we study i) the establishment time of a routine, i.e. the end time of the utterance at which the expression is established, and ii) the collaborative period of a team, i.e. the duration between first quartile (Q1) to third quartile (Q3), i.e. the interquartile range (IQR) of the establishment, where half of the establishments occur.

To study H1.1, for task performance, we check if the median establishment times are significantly earlier for better-performing teams by Spearman’s rank correlation and its statistical significance (between the median and the error). For the learning outcomes, we compare the distribution of the median establishment times of teams with positive learning outcomes (Teams 7, 9, 10, 11, 17) and others (Teams 8, 18, 20, 28, 47) by Kruskal-Wallis H test\(^\text{17}\) We consider i) the establishment times of all the routines in real time (as teams finished took different durations to complete the task), ii) the establishment times of the routines that are established in the a common time duration (of the first \(\approx 11\) mins, as this was the time taken by the quickest team), and iii) normalised establishment times, that are scaled by the duration of each team itself to reflect the ‘progress’ of a team’s interaction, from 0% progress at the beginning of the activity, to 100% when the interaction ends (either by finding an optimal solution, or by being intervened by the experimenters). To gain further insight, we compare the distribution of the establishment times (from i) to iii)) of the teams, through inspecting the box plots of all teams that are compared side-by-side and sorted by decreasing task success\(^\text{18}\).

Linking hesitation phenomena to the establishment of the routines (H1.2). To measure hesitation phenomena, we choose fillers (in particular ‘uh’ and ‘um’) specifically due to their many links with the uncertainty of the interlocutor, be it by simple hesitation (Pickett, 2018), deeper meanings of a speaker’s feeling of how knowledgeable they are (Smith and Clark, 1993), or even the listener’s impression of how knowledgeable is the speaker (Brennan and Williams, 1995). Fillers can thus be used by the interlocutor to inform the listener about upcoming new information, or even production difficulties that they are facing. Particular to the establishment of referring expressions, research has shown that disfluency (studied with the filler ‘uh’) biases listeners towards new referents (Arnold et al., 2004) rather than ones already introduced into the discourse, and helps listeners resolve reference ambiguities (Arnold et al., 2007).

To investigate H1.2, we inspect the distribution of filler times, in relation to the i) establishment and ii) priming times of routine expressions. This is done to account for when a speaker introduces a new task-specific referent into the dialogue, and when the listener makes this expression routine for the first time. In particular, we note the order of the tokens in the dialogue for the filler positions and the first token of the priming/establishment instance. Then, we check whether the distributions of filler times (by its token position) with establishment times and priming times are significantly different (by its first token’s position), by utilising a Mann-Whitney U Test, and estimate the effect size by computing Cliff’s Delta. Then, we compare these results for the teams as sorted by increasing task success: for task performance, we compute the Spearman’s correlation and its significance between Cliff’s Delta and error. For learning outcomes, we perform Kruskal-Wallis H test to compare the distribution of Cliff’s Delta values for the two groups of learning.

\(^\text{17}\)Since we have groups of 5 teams for learning, we use Kruskal-Wallis (SciPy’s implementation of Kruskal-Wallis works with \(\geq 5\) samples). This can not be used for performance.

\(^\text{18}\)Our analysis on each hypothesis is publicly available online with a script tools/7_test_the_hypotheses.ipynb among the tools, from the Zenodo Repository, DOI: 10.5072/zenodo.742549
3.4 Studying the follow up of task-specific referents with actions (RQ2)

Detecting follow-up actions of the task-specific referents. If an instruction is verbalised (for e.g. to connect “Mount Basel to Montreux”) by an interlocutor (IG), it could result in an action of connecting the two. We hence say an instruction matches an action when the instruction is executed by the other interlocutor (IF) via an action in the situated environment. We study the discrepancy created when the IF does not follow the IG, which we call mismatch of instructions-to-actions. In Example 1, the instruction (to connect Gallen to Davos) matches the action (connecting these two). The following example illustrates a dialogue excerpt that results in a mismatch:

B: Go to Mount Basel. [Instruction to add an edge to Basel.]  
A: That’s, it’s expensive. 
B: Just do it. 
A: You can’t, you can’t, I can’t because there’s a mountain there . . . 
A: So I’m going, so I’m going here.  
〈A connects Mount Interlaken to Mount Bern〉 [Mismatch!]

We extract instructions from the utterances through the interlocutors’ use of task-specific referents and find (mis)matches of instructions-to-actions. Algorithm 3, that uses Algorithm 4, accounts for partial matches of nodes, i.e. an IG need not explicitly say both node names. It builds up (“caches”) pending instructions that have not been followed, clearing them once the views are swapped. The cache of instructions is also cleared if after an edit action a match or mismatch of instructions-to-actions is detected. This allows for matching of negotiated nodes: in Example 1, Gallen-Davos was the result of a negotiation, rather than a complete given instruction by the IG. B says in one utterance “oh Mount Davos” and then in another “to Mount Gallen yeah do that”, resulting in two cached inferred instructions; (Davos,?) and (Gallen,?) respectively. This accounts for some amount of multiple speaker-turn negotiations, and possible other ways of referring to task-specific referents (e.g “Now go from there to Davos”). See Table 3 in Sec. 4.2 for another example.

Studying the temporality of the follow-up actions (H2.1). To investigate H2.1, we compare the distributions of the match and mismatch times of the teams, by following the same methodology stated for H1.1 (which investigated establishment and priming times). In particular, we check if the

---

19Within the period of a turn of views before they are swapped.  
20From Team 8, extracted/detected by using our Algorithms 1 and 3. The whole dialogue with the annotations is available in the annotated corpus (processed_data/annotated_corpus, available with the tools).  
21We design RECOGNISE-INSTRUCTIONS, described in Algorithm 1, that uses Algorithm 2, which automatically infers a sequence of instructions from an input utterance, via a rule-based algorithm. Its implementation is available with the tools.  
22We design MATCH-INSTRUCTIONS-TO-ACTIONS, described in Algorithm 3.  
23In our implementation, we combine the transcripts and the logs (from the dataset at DOI: 10.5281/zenodo.4627104) via a script (tools/5_construct_the_corpus_by_combining_transcripts_with_logs.ipynb in the tools) to generate a combined corpus (processed_data/corpus/ available with the tools). Then, we process this corpus via a script (tools/6_recognise_instructions_detect_follow-up_actions.ipynb in the tools) to generate an annotated corpus (processed_data/annotated_corpus, available with the tools). The annotated corpus contains the inferred instructions, matches, mismatches, and nonmatches if there were no instructions to be matched.  
24Though an instruction could be carried out after the views swap again, i.e. in the following turn: in our algorithm, the pending instructions are cleared at every swap, resulting in a nonmatch.
median match or mismatch times are significantly earlier for better-performing teams by Spearman’s rank correlation, and for the two groups of learning, by Kruskal-Wallis H test.

**Linking information management phenomena to follow-up actions (H2.2).** We consider the use of ‘oh’ as an information management marker, to mark a focus of speaker’s attention, which then also becomes a candidate for the listener’s attention. This creation of a joint focus of attention allows for transitions in the information state [Schiffrin 1987]. We are interested in what is commonly known by the interlocutors regarding the task (as an information state). Instructions that contain ‘oh’ in their surrounding context are used by the speaker to inform the listener about some change in their information state. This can either be done in speaker turn utterance (“Oh, this is how we do the task”) or as a backchannel (typically by the listener: “Oh okay, oh yeah . . . ”). Ultimately this increase in shared information should lead to an increase in task success.

To investigate H2.2, we check if the distributions of (mis)matches and ‘oh’ marker times are significantly different by Mann-Whitney U test, and estimate the effect size by computing Cliff’s Delta. This calculation differs from H1.2, as actions are not part of an utterance (compared to establishment time for example); hence we cannot use the order of tokens as was previously used, and instead compare the end times of the utterance that contains the marker with the (mis)matched action times. Then, as H1.2, we calculate the relation to performance and learning.

4. Results and Discussion

4.1 RQ1: How do the interlocutors use task-specific referents? Does this link to task success?

**H1.1: Task-specific referents become routine early for more successful teams.**

**For task performance.** Fig. 5 shows for each team the distribution of the establishment times of the routines. The median establishment times for all of the routines is strongly positively correlated with the task performance measure error (Spearman’s ρ = 0.69, p < .05), that is, better-performing teams establish routines earlier. We see from the figure that the results are influenced by the variation in the duration of the activity: the interaction ends for the well-performing teams when they find a correct solution, whereas the badly-performing teams continue their interaction until the experimenters intervene and stop the activity. Inspecting the distribution of the routines that are established in the common duration, i.e. first ≈ 11 mins, we see that there is a moderate correlation coefficient between the establishment times and error (ρ = 0.52, p = .15), neglecting Team 18, which established no routines at all during this period and spoke very little in this common duration. Once we normalise the establishment times by the duration for each team separately, we see that the normalised median establishment times has a positive, medium Spearman’s correlation coefficient (ρ = 0.38, p = .28), indicating a general trend that establishments occur earlier for the better-performing teams. Fig. 6 shows the normalised establishment times.

While it is expected that better-performing teams have established routines, from the figures we observe even badly-performing teams still successfully established routines. All teams, regardless of performance, have aligned to some degree. Focusing on Fig. 5 the establishment tends to occur

---

25 The code that reproduces all the results and figures given in this paper are available from the Zenodo Repository, DOI: 10.5072/zenodo.742549

26 The magnitude of Spearman’s correlation coefficient (ρ) can be interpreted by using the thresholds e.g. from Evans (1996), i.e. 0.00 – 0.19 “very weak”, 0.20 – 0.39 “weak”, 0.40 – 0.59 “moderate”, 0.60 – 0.79 “strong”, and 0.80 – 1.00 “very strong”.
Figure 5: The team’s establishment times for H1.1. The teams are sorted by decreasing task performance (i.e. increasing error). The boxplots, with the same colour have the same error (sorted by increasing duration for the ties). The thick, bold boxplots with whiskers with maximum 1.5 IQR show the distributions of the establishments that occurred in the common duration (i.e. as marked by the dashed green line), while the thin boxplots show the distributions through the total duration of interaction. The learning outcome of each team is indicated with a plus (‘+’) for learn > 0, or a minus (‘−’) otherwise. Solid lines indicate the end of the interaction, by submitting a correct solution (in green) or timing out (in red). The thin blue lines indicate submission of a solution, with the number showing the error quantifying how far the submitted solution is from an optimal solution in terms of its cost (e.g. error = 0% means the team has found an optimal solution). The red and blue dots indicate the utterance times of the interlocutors, to give an idea of when the interlocutors are speaking versus when are their establishment times.
Figure 6: The teams’ establishment times as normalised by the duration for each team separately, i.e. 100% indicates the end of that team’s interaction. Sorting is as in Fig. 5.
around the middle of the dialogue: establishment times have mean of the medians = 63.0% (combined $SD = 22.2\%$). While we hypothesised that establishment will happen early in the dialogue, this is the case for an ideal dialogue; people will ‘share’ expressions earlier. However, we observe that there is an exploratory period (roughly the period before Q1), where the interlocutors take the time to understand the task, followed by a collaborative period that corresponds to the establishment period (roughly the period between Q1 and Q3). We expect that if the interlocutors had to complete the task again, the establishment/collaborative period would be closer to the start of the dialogue.

We observe that five teams (7, 8, 10, 18, and 47) that could not find a correct solution started collaborating later in their dialogue, in terms of when they establish most of their routines (median establishment time > 60%). Though the measures of task performance would only reflect that these teams performed badly with a final overall score of performance, our alignment measures reflect details of the performance. Interlocutors are gently reminded a few minutes before the end of the task the remaining time, but were not rushed to find a solution: which could have changed the way they aligned by forcing an establishment period. Thus, we observe through our alignment measures that badly performing teams were simply slower to collaborate and establish routines.

Two teams have established routines later in their dialogue: Team 17 that found a correct solution, and Team 18 that could not (their median establishment times are around 70%). Yet, 17 has a focused establishment period (with a smaller IQR = 18% vs. 30% of the time, respectively, see Fig. 6). We interpret that Team 17 was able to turn it around and find a correct solution, while Team 18 did not; ending up as the worst performing team. 17 has positive learning gain ($learn = 25\%$), while it is negative for Team 18 ($learn = -56\%$, the highest decrease among the transcribed teams, see Fig. 3). While bad performance could be reflected through a collaborative period starting later in their dialogue (such as Team 18; with later establishment, and not finding a solution in time), Team 17 shows that there are exceptions to this. Team 18 also possibly got confused, reflected in the high and negative $learn$.

The collaborative and exploratory periods can also be observed by examining the error progression of teams’ submitted solutions during the task, as shown in Fig. 6. During the exploratory period, Nine out of ten teams had their highest error. For teams that performed well, their collaborative period had their lowest error before finding an optimal solution (thus ending the task), and for other teams; their closest solution to the optimal solution. There are 8 such teams that exhibit this behaviour. The teams that did not achieve a correct solution, never regressed to their largest error from the exploratory period. Looking at the number of attempts, for the well-performing teams, their collaborative period was productively used, with their next solution reaching an optimal cost (Teams 17 and 28), or one more attempt before their optimal cost (Team 20). Several teams that did not perform well have a greater number of attempts submitted after their collaborative period (Teams 9, 7, 8, 10, 47). Their submission pattern of attempts indicate a “trial-and-error” strategy on how to solve the task, for example, Teams 8 and 9 increasing their submissions after their collaborative period, or Teams 7, 10 and 47 submitting throughout.

For learning outcomes. The distribution of the median establishment times are not statistically significantly different for teams with positive learning outcomes (Teams 7, 9, 10, 11, 17) vs. others (Teams 8, 18, 20, 28, 47) by Kruskal-Wallis H test ($H = 0.88, p = .35$ for times in absolute values, and $H = 0.10, p = .75$ for normalised times).

Synopsis. Overall, the results support H1.1 for task performance, while they are inconclusive for learning outcomes. Surprisingly, we see that all teams, regardless of task success, are (lexically)
aligned to some degree. Thus, inspecting the distributions of the establishment times together with their proposed solutions gave further insight into the collaboration processes, specifically revealing a “collaborative period” that most of the teams came up with their best solutions. This captures some patterns of alignment that might have otherwise been overlooked.

**H1.2: Hesitation phenomena are more likely to occur in the vicinity of priming and establishment of task-specific referents for more successful teams.**

We consider hesitation phenomena as cued by the presence of a filler, and thus investigate how are the fillers distributed as compared to the priming and establishment of the routines. The results for Mann-Whitney U test and effect size as estimated by Cliff’s Delta ($\delta$) are given in Table 2. $\delta$ ranges from $-1$ to $1$, where $0$ would indicate that the group distributions overlap completely; whereas values of $-1$ and $1$ indicate a complete absence of overlap with the groups. For example, $-1$ indicates that all fillers occur before priming times, and $1$ indicates that all fillers occur after priming.

We observe that filler and **priming times** differ significantly for most of the teams (for eight out of ten teams by Mann-Whitney U test $p < .05$). We see positive $\delta$ values, except for one team. Most teams thus have fillers that occur after priming times. We see that the magnitude of $\delta$ for most teams is large except for Team 7 (excluding Teams 20 and 8 that are not statistically significant$^{27}$). We interpret that most fillers do occur visibly after priming times (with little overlap from the large effect sizes). We observe that filler and **establishment times** differ significantly for most of the teams (for seven out of ten teams by Mann-Whitney U test $p < .05$). We see $\delta$ ranging from $-0.67$ for Team 7, to $0.33$ for Team 11 (though most are negative). However, we see that the magnitude of $\delta$ for most teams is small, with the exception of two teams. This means that the distributions of fillers differ with a small effect size compared to the distributions of establishment times, especially considering that the token number values do not overlap; i.e. we do not expect an effect size of $0$. We thus interpret most fillers do occur around (from positive and negative $\delta$ values) establishment times (with larger overlap from small effect sizes).

These results indicate that in the process of routine formation, in between the priming and the establishment of the expression, there seems to be a period in which the interlocutors exhibit hesitation phenomena. This placement of fillers is of interest due to the potentially unfamiliar vocabulary of the task-specific referents that the interlocutors had to utilise in the situated environment. The results demonstrate a lack of hesitation phenomena at the start of the formation of a routine; i.e. they occur visibly after the priming of expressions that contain task-specific referents. The large effect size shows that this is predominantly the case for most teams. In addition to this, fillers were found to occur around establishment times. We suggest that a part of establishment is often a clarification request, as shown by the following example. Red indicates the routine expression, while blue indicates the filler.

$^{27}$The magnitude of Cliff’s Delta ($\delta$) can be interpreted by using the thresholds from Romano et al. (2006), i.e. $|\delta| < 0.147$ “negligible”, $|\delta| < 0.33$ “small”, $|\delta| < 0.474$ “medium”, and otherwise “large”.

20
Table 2: The results for the Mann-Whitney U test that compares the distribution of filler times with establishment and priming times for H1.2. The effect size is estimated by Cliff’s Delta ($\delta$). Teams are sorted by decreasing task performance i.e. increasing error. The horizontal line separates well-performing teams (that found a correct solution) from badly-performing teams. $U$ is the U statistic, and $p$ is the 'two-sided' p-value of a Mann-Whitney U test (without continuity correction as there can be no ties, via our unique token number assignment).

<table>
<thead>
<tr>
<th>Team</th>
<th>count</th>
<th>filler routine</th>
<th>filler median (%)</th>
<th>priming</th>
<th>establishment</th>
<th>$U$</th>
<th>$p$</th>
<th>$\delta$</th>
<th>$U$</th>
<th>$p$</th>
<th>$\delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>38</td>
<td>58</td>
<td>11.6</td>
<td>3.5</td>
<td>15.2</td>
<td>1600.5</td>
<td>&lt; .05</td>
<td>0.45</td>
<td>828.5</td>
<td>&lt; .05</td>
<td>-0.25</td>
</tr>
<tr>
<td>17</td>
<td>27</td>
<td>41</td>
<td>14.0</td>
<td>6.2</td>
<td>17.1</td>
<td>823.0</td>
<td>&lt; .05</td>
<td>0.49</td>
<td>431.0</td>
<td>.12</td>
<td>-0.22</td>
</tr>
<tr>
<td>20</td>
<td>59</td>
<td>51</td>
<td>20.7</td>
<td>18.1</td>
<td>22.8</td>
<td>1746.5</td>
<td>.15</td>
<td>0.16</td>
<td>1139.5</td>
<td>&lt; .05</td>
<td>-0.24</td>
</tr>
<tr>
<td>11</td>
<td>35</td>
<td>70</td>
<td>34.5</td>
<td>3.5</td>
<td>24.4</td>
<td>2279.0</td>
<td>&lt; .05</td>
<td>0.82</td>
<td>1624.0</td>
<td>&lt; .05</td>
<td>0.33</td>
</tr>
<tr>
<td>9</td>
<td>81</td>
<td>131</td>
<td>46.3</td>
<td>11.4</td>
<td>37.4</td>
<td>8805.5</td>
<td>&lt; .05</td>
<td>0.66</td>
<td>6561.5</td>
<td>&lt; .05</td>
<td>0.24</td>
</tr>
<tr>
<td>47</td>
<td>56</td>
<td>74</td>
<td>19.5</td>
<td>6.7</td>
<td>21.9</td>
<td>3189.0</td>
<td>&lt; .05</td>
<td>0.54</td>
<td>1627.0</td>
<td>&lt; .05</td>
<td>-0.21</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>65</td>
<td>20.1</td>
<td>8.7</td>
<td>20.4</td>
<td>982.0</td>
<td>&lt; .05</td>
<td>0.51</td>
<td>660.0</td>
<td>.92</td>
<td>0.02</td>
</tr>
<tr>
<td>8</td>
<td>26</td>
<td>62</td>
<td>19.8</td>
<td>15.7</td>
<td>36.0</td>
<td>883.0</td>
<td>.48</td>
<td>0.10</td>
<td>461.0</td>
<td>&lt; .05</td>
<td>-0.43</td>
</tr>
<tr>
<td>7</td>
<td>27</td>
<td>59</td>
<td>24.9</td>
<td>25.9</td>
<td>37.8</td>
<td>562.0</td>
<td>&lt; .05</td>
<td>-0.29</td>
<td>259.0</td>
<td>&lt; .05</td>
<td>-0.67</td>
</tr>
<tr>
<td>18</td>
<td>52</td>
<td>57</td>
<td>10.8</td>
<td>4.3</td>
<td>11.8</td>
<td>2136.0</td>
<td>&lt; .05</td>
<td>0.44</td>
<td>1326.0</td>
<td>.34</td>
<td>-0.11</td>
</tr>
</tbody>
</table>

A: *Mount Zurich to Mount Bern.*

......

B: *uh* isn’t already *Mount Zurich to Mount Bern* isn’t it connected?

......

(4)

A: So if we erase *Zurich* or Mount Ber- to Mount Bern or Mount *Zurich* to Mount Gallen?

B: Wait *uh Zurich* to Mount ... ?

Speakers are able to prime these expressions without exhibiting this hesitation phenomena, but this does not guarantee that the primed expression was fully understood by the listener, as shown by the results of the Mann-Whitney U test. Following the idea of IF and IG pairs in the dialogue, the IG (speaker) tends to be in the abstract view, and can see the minimally represented names of gold mines. They need only concentrate on the gold mines that have the lowest cost to connect. Perhaps the reason why the IG does not exhibit (by our measures) this uncertainty is because they can see the task-specific referents available to them written down on the map. Expressions that contain task-specific referents could successfully become part of the IG’s expression lexicon when given these new referents. The IG may also feel at ease to read out these expressions. The IF (listener) must follow the instructions with actions, and since they can not see the cost of adding and removing edges, they must search for the specific gold mine names given by the IG. This could create an uncertainty in the IG, and bring about a need to clarify.

For task performance. The distribution of the Cliff’s Delta between filler times and both priming or establishment times has a very weak correlation coefficient with error (Spearman’s $\rho = -0.18$, $p = .62$ for priming, $\rho = 0.06$, $p = .88$ for establishment). Therefore, we can not conclude that there
is a significant relationship between how early are the priming or establishment times, and how well a team performs.

**For learning outcomes.** The distribution of the Cliff’s Delta that quantifies how different are filler times from priming or establishment times are not statistically significantly different for teams with positive learning outcomes (Teams 7, 9, 10, 11, 17) vs. others (Teams 8, 18, 20, 28, 47) by Kruskal-Wallis H tests ($H = 1.32$, $p = .25$ for both). Therefore, we can not conclude that a significant difference exists, between how early fillers occur compared to priming (or establishments) between these two learning groups.

**Synopsis.** Regarding H1.2, we can not conclude that the usage of fillers occur in the vicinity of priming or establishments times for more successful teams (for neither better-performing teams, nor higher-learning teams). Overall, we see that for most of the teams, the fillers tend to occur visibly after priming times, and around establishment times. We still believe that this gives us a fascinating insight into the way the interlocutors use task-specific expressions in a situated environment.

### 4.2 RQ2: How do the interlocutors follow up the use of task-specific referents with actions? Does this link to task success?

**H2.1: Instructions are more likely to be followed by a corresponding action early in the dialogue for more successful teams.**

To investigate H2.1, we automatically infer (mis)matches\(^{28}\) where a verbal instructions’ follow-up action is labelled as a (mis)match following the methodology described in Sec. [3.3](#). Then, we inspect the distribution of the (mis)matches of the teams, and see how this related to task success.

**For task performance.** The median match times has a moderate positive correlation coefficient with the task performance measure error (Spearman’s $\rho = 0.59$, $p = .08$), that is, better-performing teams have matches earlier. This follows the results in H1.1, regarding establishment times. For the matches in the common duration (i.e. first $\approx 11$ mins), there is a moderate correlation coefficient ($\rho = 0.45$, $p = .23$, ignoring Team 18 that has no matches in this common period). Fig. [7](#) shows the the distribution of the (mis)match times for each team. When the times are normalised by the duration for each team separately, we observe a positive, moderate Spearman’s correlation coefficient ($\rho = 0.54$, $p = .11$), that indicates a general trend that matches happen earlier for the better-performing teams.

Fig. [8](#) shows for each team, the distribution of the match and mismatch times, normalised by the duration of each team itself. While it is natural to think that instructions will be followed by a match for more successful teams in a structured and organised manner, we observe that teams that performed badly and teams that did not learn also have a certain period of matches. Therefore, all teams, independent of their task success, have their IG’s instructions matched to the IF’s actions to some extent. Match times have mean of the medians = 62.6% (combined $SD = 21.9$%), and mismatches have mean of medians = 67.6% (combined $SD = 26.0$%).

By inspecting the outputs of the automatically inferred instructions in Table [3](#), occasionally, we see that the traditional roles of IF and IG are not maintained, even for the brief fixed time of views

---

\(^{28}\)As described in Algorithm [3](#). We process this combined dialogue and actions corpus via a script (tools/6_recognise_instructions_detect_follow-up_actions.ipynb in the tools) to generate an annotated corpus (processed_data/annotated_corpus, available with the tools). The annotated corpus contains the inferred instructions, their pending lists used for matching, matches, mismatches, and nonmatches if there were no instructions to be (mis)matched.
Table 3: An example from the output of recognising instructions and detecting follow-up actions (by Algorithms 1 and 3), from Team 10. View denotes which view the interlocutor is in (refer to Sec. 2.1). Annotations denotes the automatically inferred instructions and follow-up actions in the activity. For example, InstructA indicates that interlocutor A has given an instruction to add two nodes (inferred from referents), which can be partially recognised (Gallen, ?). As shown, the algorithm builds up (or “caches”) instructions until an edit action is performed (‘-‘) in Utt. no.). Note, since B is in the visual view his/her inferred instruction is deliberately not matched.

<table>
<thead>
<tr>
<th>Utt. no.</th>
<th>View</th>
<th>Verb</th>
<th>Utterance</th>
<th>Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 198</td>
<td>Abstract</td>
<td>says</td>
<td>Maybe we start from, Mount Zermatt ?</td>
<td>InstructA(Add(Zermatt,?))</td>
</tr>
<tr>
<td>B 199</td>
<td>Visual</td>
<td>says</td>
<td>No lets do Mount Davos to, where do you wanna go?</td>
<td>InstructA(Add(Zermatt,?), InstructB(Add(Davos,?))</td>
</tr>
<tr>
<td>A 200</td>
<td>Abstract</td>
<td>says</td>
<td>. . . to Mount, St Gallen.</td>
<td>InstructA(Add(Zermatt,?), InstructB(Add(Davos,?), InstructA(Add(Gallen,?)))</td>
</tr>
<tr>
<td>B 201</td>
<td>Visual</td>
<td>says</td>
<td>Okay.</td>
<td>As previous</td>
</tr>
<tr>
<td>B -</td>
<td>Visual</td>
<td>adds</td>
<td>Gallen-Davos</td>
<td>InstructA(Add(Zermatt,?), MatchB(InstructA(Add(Gallen,?))))</td>
</tr>
</tbody>
</table>

(i.e. within a turn). The IF, having just been switched from IG, may have their own ideas about the next best edit action, as seen in Utt. no. 199 (prior to Utt. no. 198, the views were swapped, with B being in the abstract view, meaning B was just previously the IG). Thus the interlocutors can also conduct a negotiation where they collaboratively decide which action to take. It is therefore not always the case that the IG is in the abstract view deciding and instructing what the IF in the visual view should do; and this is supported by the frequent swapping of views.

For learning outcomes. By inspecting the median of the match times with learn, we observe that they are not statistically significantly different for teams with positive learning outcomes (Teams 7, 9, 10, 11, 17) vs. others (Teams 8, 18, 20, 28, 47) by Kruskal-Wallis H tests ($H = 0.54$, $p = .47$ for times in absolute values, and $H = 0.10$, $p = .75$ for normalised times).

Though not statistically significant by these measures, from the Algorithm 3’s results, we see that all teams have more inferred matches than mismatches (see Table 4 for reference). Teams 18 and 20 that learnt the least, have the highest match-to-mismatch ratio (3.6 and 4.0 matches of instructions-to-actions for every 1 mismatch respectively.), as well as Team 7 (3.3). Teams 8 and 47, which are also amongst the teams that had a negative learning gain (but close to zero, substantially higher than 18 and 20), have lower match-to-mismatch ratio of 2.7 and 1.3 respectively. We observe that Teams 10, 11 and 17, who had the highest learning gain, comparatively have a lower ratio of matches-to-mismatches of 2.4, 1.8 and 2.3 respectively. We interpret that it is important for learning that interlocutors have a good ratio of matches-to-mismatches, and not just be in total (blind) agreement with the other. This is consistent with previous research, as stated in Sec. 3.1. By just inspecting performance, superficially, Team 20 performed well. However, they did not learn (in fact, “unlearnt”, as shown with a negative learning outcome). Observing their progression of costs for submitted solutions (from 191%, to 109%, to finding an optimal solution) informs us that they “got lucky” in finding a solution, however did not learn, as is further evidenced by only a few, and sparse mismatched instructions-to-actions. Indeed, before finding an optimal solution, their previous error (109%) was substantially higher than any other team that performed badly.
Figure 7: The team’s match and mismatch times in blue (‘+’) and red (‘×’) respectively, to study H2.1. The teams are sorted by decreasing learning outcomes (i.e. decreasing learn). See Fig. 5 for a description of the green, red and blue lines.
Figure 8: The teams’ match and mismatch times in green and red respectively, as normalised by the duration for each team separately. Sorting is as in Fig. 7.
From the results, we interpret that positive learning teams seem to have conflict(s) that they resolved collaboratively while building and submitting solutions (and receiving feedback from the robot), whereas the others either had less or unresolved conflicts. By inspecting how the (mis)matches are distributed, and how they are they positioned in relation to the costs of the submitted solutions, we observe nuances of conflicts and their resolutions. For instance, consider a high-learning Team 11. The team began with a high cost (191%, i.e. they basically connected everything to each other, with many redundant connections). One interpretation is that they initially misunderstood the goal of the task of connecting the graph minimally. They had many mismatches (“a mismatch period”), which is followed by many matches ending with (at Q3 of matches) their best solution, getting very close (5%, i.e. they did not notice that they could replace a particular connection with a better/lower-cost one). Although the team could not find a correct solution, they learnt a lot from this interaction flow. In comparison, Team 20 also started with that high cost of 191%. Yet, they could not resolve the conflict, as their (mis)matches did not result in them getting closer to an optimal solution, but rather still keeping to high cost solutions, and repeating these high cost submissions (109%, i.e. more than double the cost they could make with many redundant connections). Although they found a correct solution next, since they did not have the conflict resolution period that could be cued by (mis)matches, they did not end up learning.

**Synopsis.** We see that all teams have (mis)matches, irrespective of their task success; therefore, it is not necessary to follow-up with an action in order to succeed. In terms of task performance, we observe a general trend that supports the H2.1, i.e. better performing teams tend to follow up the instructions with actions (as matches) early overall (the absolute times), as well as in their dialogues (normalised times). For learning, although the general trend is not statistically confirmed, we gained insight into the nuances of the dynamics of interaction: by inspecting the normalised (mis)match time plots alongside with the submission costs. There seems to be conflicts that may be collaboratively resolved and resulted in learning, or may not be resolved and have adverse effects on the learning outcomes.

**H2.2: When instructions are followed by a corresponding or a different action, the action is more likely to be in the vicinity of information management phenomena for more successful teams.**

To investigate H2.2, we consider the distribution of the information management marker ‘oh’ through time, in relation to the distribution of matches and mismatches. In particular, we check if the distributions of ‘oh’ times and (mis)matches are significantly different by Mann-Whitney U test, and estimate the effect size by computing Cliff’s Delta ($\delta$). The results of the tests for each team are given in Table 4. Here, $\delta = -1$ would mean that all ‘oh’s occur earlier than (mis)match times, and 1; that all ‘oh’s occur later. For all teams except one, the value of $\delta$ is negative, indicating that the distribution of ‘oh’ tends to occur earlier than (mis)match times. We observe that ‘oh’ and (mis)match times do not significantly differ for half of the teams (Mann-Whitney U test $p < .05$).

---

29 And are not only exploring the graph to see the costs and available connections, as is evidenced by their following submissions that had again a very high cost (127%), only later decreasing further through several submissions.

30 We note that the ‘oh’ marker occurs 333 times in the transcripts (average per transcript = 33.3, SD = 20.3). See Table for the number of utterances that contain one or more ‘oh’s for each team.

31 As H1.2, $\delta$ ranges from $-1$ to 1, where 0 would mean that the group distributions overlap completely; whereas values of $-1$ and 1 indicate a absence of overlap between the groups.
Table 4: The results for Mann-Whitney U tests that compare the distribution of information marker (i.e. ‘oh’) times, with (mis)match times for H2.2. The effect size estimated by Cliff’s Delta ($\delta$). Teams are sorted by decreasing task performance, i.e. increasing error. U is the U statistic, and $p$ is the ‘two-sided’ $p$-value of a Mann-Whitney U test (without continuity correction).

<table>
<thead>
<tr>
<th>Team</th>
<th>oh</th>
<th>match (number of utterances)</th>
<th>mismatch</th>
<th>match/mism.</th>
<th>median (%)</th>
<th>oh</th>
<th>match</th>
<th>mismatch</th>
<th>(mis)match</th>
<th>$U$</th>
<th>$p$</th>
<th>$\delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>24</td>
<td>19</td>
<td>13</td>
<td>1.5</td>
<td>61.7</td>
<td>62.7</td>
<td>73.6</td>
<td>344.0</td>
<td>.51</td>
<td>-0.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>20</td>
<td>23</td>
<td>10</td>
<td>2.3</td>
<td>61.1</td>
<td>61.1</td>
<td>69.4</td>
<td>318.0</td>
<td>.83</td>
<td>-0.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>65</td>
<td>24</td>
<td>6</td>
<td>4.0</td>
<td>35.0</td>
<td>58.0</td>
<td>42.3</td>
<td>715.0</td>
<td>&lt;.05</td>
<td>-0.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>15</td>
<td>34</td>
<td>18</td>
<td>1.8</td>
<td>57.3</td>
<td>59.5</td>
<td>38.5</td>
<td>427.0</td>
<td>.58</td>
<td>0.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>29</td>
<td>28</td>
<td>9</td>
<td>3.1</td>
<td>42.9</td>
<td>59.9</td>
<td>70.2</td>
<td>429.0</td>
<td>.16</td>
<td>-0.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>29</td>
<td>28</td>
<td>21</td>
<td>1.3</td>
<td>53.4</td>
<td>56.8</td>
<td>75.7</td>
<td>488.0</td>
<td>&lt;.05</td>
<td>-0.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>18</td>
<td>36</td>
<td>15</td>
<td>2.4</td>
<td>47.6</td>
<td>67.3</td>
<td>65.3</td>
<td>264.0</td>
<td>&lt;.05</td>
<td>-0.42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>55</td>
<td>30</td>
<td>11</td>
<td>2.7</td>
<td>36.3</td>
<td>65.3</td>
<td>86.3</td>
<td>522.0</td>
<td>&lt;.05</td>
<td>-0.54</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>51</td>
<td>43</td>
<td>13</td>
<td>3.3</td>
<td>49.7</td>
<td>65.5</td>
<td>79.7</td>
<td>841.0</td>
<td>&lt;.05</td>
<td>-0.41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>25</td>
<td>7</td>
<td>3.6</td>
<td>69.6</td>
<td>69.5</td>
<td>75.0</td>
<td>157.0</td>
<td>.36</td>
<td>-0.18</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We see $\delta$ ranging from −0.54 for Team 8, to 0.09 for Team 11: this indicates that ‘oh’ occurs differently compared to the (mis)match times within the teams. We see that for the teams that had significantly different distributions, the effect size ranges from negligible ($|\delta| < 0.147$) to medium ($|\delta| < 0.474$), with the exception of Team 8, that has large effect size ($\delta = -0.54$).

**For task performance.** Since the $\delta$ values are varying from being around 0 to negative values (see Table 4), for all teams except one, oh’s occur earlier than (mis)match times. Following this, the $\delta$ values between ‘oh’ times and (mis)match times has a medium negative correlation coefficient with error (Spearman’s $\rho = -0.53, p = .12$). This means that the better the performance, the more the overlap between (mis)match times and ‘oh’s. There is a general trend that for well-performing teams the ‘oh’s occur more in the vicinity of (mis)match times with larger overlap between the two groups, and the badly-performing teams have comparably less overlap. Thus the results support H2.2.

**For learning outcomes.** The distribution of the Cliff’s Delta, that quantifies how different ‘oh’ times are from (mis)match times, is not statistically significantly different for teams with positive learning outcomes (Teams 7, 9, 10, 11, 17) vs. others (Teams 8, 18, 20, 28, 47) by Kruskal-Wallis H test ($H = 0.27, p = .60$).

We see from Table 4 there is a delicate balance between performance and learning. Team 20 for example, performed well but learnt nothing, and differ with statistical significance in their distribution of ‘oh’ versus (mis)match times compared to other better performing teams. For comparison, we thus choose Team 17, who had high task success (i.e. performed and learnt well), vs. Team 20, who performed well but did not learn, and then we sample utterances from these two teams at the start and end of their dialogue. See Table 5 and Table 6 for excerpts from Teams 20 and 17, respectively. For both teams, we see the first occurrences of ‘oh’ in the tables (corresponding to the exploratory period, from H1.1) and a period of nonmatched instructions for both teams, as they individually figure out the constraints of the task and navigate the situated environment. Keeping
Table 5: Excerpts that contain information management marker ‘oh’ from Team 20, who performed well but did not learn. Annotations could have the pending instructions, and (mis)matched or nonmatched actions. Note, Nonmatch\textsubscript{Interlocutor} frequently occurs when the interlocutor follows their own intentions, as seen after Utterance 10, Nonmatch\textsubscript{A}(Do\textsubscript{A}(Add)), where A connects two nodes (for brevity, nodes given in Utterance column, unless partial (mis)match).

<table>
<thead>
<tr>
<th>Int.</th>
<th>Utl.</th>
<th>Verb</th>
<th>Utterance</th>
<th>Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>10</td>
<td>says</td>
<td>I’m just gonna . . .</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>adds</td>
<td>Luzern-Zermatt</td>
<td>Nonmatch\textsubscript{A}(Do\textsubscript{A}(Add))</td>
</tr>
<tr>
<td>A</td>
<td>11</td>
<td>says</td>
<td>Uh . . .</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>12</td>
<td>says</td>
<td>Uh . . .</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>13</td>
<td>says</td>
<td>Oh there.</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>14</td>
<td>says</td>
<td>Oh two three.</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>15</td>
<td>says</td>
<td>Oh that’s what you’ve been doing this all time.</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>56</td>
<td>says</td>
<td>Oh I think we have to connect all of them.</td>
<td>Instruct\textsubscript{A}(Add(Gallen,?))</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>adds</td>
<td>Luzern-Interlaken</td>
<td>Mismatch\textsubscript{B}(Instruct\textsubscript{A}(Add(Gallen,?)))</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>adds</td>
<td>Luzern-Zurich</td>
<td>Nonmatch\textsubscript{B}(Do\textsubscript{B}(Add))</td>
</tr>
<tr>
<td>A</td>
<td>57</td>
<td>says</td>
<td>Oh.</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>58</td>
<td>says</td>
<td>Okay I did some</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>59</td>
<td></td>
<td>okay for me.</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>adds</td>
<td>Luzern-Davos</td>
<td>Nonmatch\textsubscript{A}(Do\textsubscript{A}(Add))</td>
</tr>
<tr>
<td>A</td>
<td>60</td>
<td>says</td>
<td>Oh no.</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>61</td>
<td>says</td>
<td>I think we are doing terrible.</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>450</td>
<td>says</td>
<td>Oh.</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>451</td>
<td>says</td>
<td>3.</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>452</td>
<td></td>
<td>What?</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>453</td>
<td>says</td>
<td>Let me . . .</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>removes</td>
<td>Luzern-Zermatt</td>
<td>Nonmatch\textsubscript{A}(Do\textsubscript{A}(Remove))</td>
</tr>
<tr>
<td>A</td>
<td>454</td>
<td>says</td>
<td>There you go.</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>455</td>
<td></td>
<td>What no.</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>456</td>
<td>says</td>
<td>You are erasing my mistake.</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>457</td>
<td></td>
<td>How dare you.</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>458</td>
<td>says</td>
<td>I know.</td>
<td>-</td>
</tr>
<tr>
<td>459</td>
<td></td>
<td></td>
<td>Wait, what?</td>
<td>-</td>
</tr>
<tr>
<td>460</td>
<td></td>
<td></td>
<td>Can I get pencil again?</td>
<td>-</td>
</tr>
<tr>
<td>461</td>
<td></td>
<td></td>
<td>Oh oh.</td>
<td>-</td>
</tr>
<tr>
<td>462</td>
<td></td>
<td></td>
<td>Oh.</td>
<td>-</td>
</tr>
<tr>
<td>463</td>
<td></td>
<td></td>
<td>Okay.</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>464</td>
<td>says</td>
<td>We messed up again didn’t we?</td>
<td>-</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>

28
Table 6: Excerpts that contain information management marker ‘oh’ from Team 17, who had high task success (i.e. performed and learnt well). Utt. stands for the utterance number, which is not applicable for the edit actions. See the caption of Table 5 for further details.

<table>
<thead>
<tr>
<th>Int.</th>
<th>Utt.</th>
<th>Verb</th>
<th>Utterance</th>
<th>Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>I</td>
<td>4</td>
<td>says</td>
<td>So you only build from something that is already connected.</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>says</td>
<td>Oh.</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>6</td>
<td>says</td>
<td>Oh okay.</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>adds</td>
<td>Zermatt-Davos</td>
<td>Nonmatch$_B$(Do$_B$(Add))</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>adds</td>
<td>Gallen-Davos</td>
<td>Nonmatch$_B$(Do$_B$(Add))</td>
</tr>
<tr>
<td>A</td>
<td>-</td>
<td>adds</td>
<td>Zurich-Davos</td>
<td>Nonmatch$_A$(Do$_A$(Add))</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>adds</td>
<td>Basel-Bern</td>
<td>Match$_B$(Instruct$_A$(Add(Basel,?)))</td>
</tr>
<tr>
<td>A</td>
<td>61</td>
<td>says</td>
<td>Yeah, and then go to Mount Zurich.</td>
<td>Instruct$_A$(Add(Zurich,?))</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>adds</td>
<td>Basel-Zurich</td>
<td>Match$_B$(Instruct$_A$(Add(Zurich,?)))</td>
</tr>
<tr>
<td>A</td>
<td>62</td>
<td>says</td>
<td>Yeah.</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>63</td>
<td>says</td>
<td>Oh no that costs more.</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>64</td>
<td>says</td>
<td>uh ...</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>65</td>
<td>says</td>
<td>We should erase it.</td>
<td>-</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>A</td>
<td>266</td>
<td>says</td>
<td>Then do Mount Bern to Mount Zermatt.</td>
<td>Instruct$_A$(Add(Bern, Zermatt))</td>
</tr>
<tr>
<td>A</td>
<td>267</td>
<td>says</td>
<td>Maybe that’s better.</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>268</td>
<td>says</td>
<td>You can’t do that.</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>269</td>
<td>says</td>
<td>Oh.</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>270</td>
<td>says</td>
<td>Then do ...</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>271</td>
<td>says</td>
<td>Mount Bern to Mount Interlaken?</td>
<td>Instruct$_B$(Add(Bern, Interlaken))</td>
</tr>
<tr>
<td>A</td>
<td>272</td>
<td>says</td>
<td>Yeah.</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>273</td>
<td>says</td>
<td>I think that’s 4 though.</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>adds</td>
<td>Interlaken-Bern</td>
<td>Mismatch$_B$(Instruct$_A$(Add(Bern, Zermatt)))</td>
</tr>
<tr>
<td>A</td>
<td>274</td>
<td>says</td>
<td>So don’t do that.</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>275</td>
<td>says</td>
<td>Is that 4?</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>276</td>
<td>says</td>
<td>Oh yeah it’s, it is 4.</td>
<td>-</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
in mind the stylistic differences amongst interlocutors (for example, in Team 20 we clearly see that interlocutor A tends to use ‘oh’ frequently), we still see interesting differences between the teams in terms of their usage of ‘oh’s and their instructions-to-actions.

In Team 20 (Utt. no. 56 − 61), we see both interlocutors using ‘oh’ traditionally as an information management marker (e.g. “Oh I think we have to connect all of them”); but from the mismatch/nonmatch of instructions-to-actions, we see that this information state of the interlocutor is not transferred to the collective information state of both interlocutors. Essentially, the interlocutors are working in isolation, individually gaining (perceived) information about the situated environment (measured by use of ‘oh’), and then following up with their own intentions in isolation (e.g. mismatch and nonmatch between utt. no. 56 and 57). To contrast, we clearly see that Team 17 (Utt. no. 61 − 65) has this transition of information from the interlocutor to the listener with their use of ‘oh’, signalling information in a shared focus of attention (with A saying “Oh no that costs more” and B responding “We should erase it”). We see similar patterns of behaviour towards the end of the dialogue (by this point, interlocutors have had opportunity to build a collaboration with each other).

The intuition behind H2.2 seems reasonable here; Team 20 for example, had 65 occurrences of the information marker ‘oh’, but they significantly differ from the 30 times the instructions given by the IG was (mis)matched by the IF. Interestingly, we observe in the last period for Team 20 a lack of inferred instructions. Inferred instructions are a precursor to an action being (mis)matched or nonmatched. Indeed, there is also a lack of nonmatched actions, which typically occur when the interlocutor takes the liberty to follow their own intentions. Here, they do not even verbalise their own intentions, visible from the lack of inferred instructions. This indicates a strong tendency of working in isolation despite the design of the task.

Synopsis. Overall, regardless of a team’s task success, we see that ‘oh’ does occur earlier than (mis)matched instructions-to-actions, as evidenced by the negative effect sizes of Cliff’s Delta. Specifically for performance, there is a general trend that for well-performing teams the ‘oh’s occur more in the vicinity of (mis)match times with larger overlap between the two groups, and the badly-performing teams have comparably less overlap. This result supports H2.2. While we cannot conclude on the results for learning, using our automatically annotated instructions-to-actions and occurrences of ‘oh’, we still gain some insight into the way the teams collaborate.

5. Conclusion

In this article, we are interested in how children collaborate as they solve a problem together, in which what they say and do is strongly tied to how they perform, and subsequently what they will ultimately learn from the situated activity. To investigate this relationship, we consider the corpus of data (dialogue transcripts from audio files and action logs) generated by teams of two children engaged in a collaborative learning activity, called JUSThink, which aims at providing an intuitive understanding of graphs and spanning trees. Collaborative learning activities are a particularly interesting type of collaborative task, due to their “multi-layered goals”, typically including immediate, performance goals (e.g., finding the solution to a math problem) and deeper, learning goals (e.g., understanding the notion of equation).

---

32 Future work could expand more on the inferred instructions and nonmatches and their implications of collaboration, rather than only focusing on (mis)matched actions.
Collaboration often involves a dialogue amongst interlocutors. It has been shown that a dialogue is successful when there is alignment between the interlocutors, at different linguistic levels. However, research on alignment in dialogue shows a lack of (automatic) measures capable of capturing the links between alignment and task success, particularly for collaborative learning settings.

This work builds upon automatic state-of-the-art alignment measures to study a collaborative learning activity eliciting completely spontaneous spoken dialogues. We introduce new measures that allow us to study another level of alignment in situated dialogues; i.e. via automatically inferring instructions, and then linking those instructions to (matched or mismatched) actions. Specifically we consider in the situated environment the interlocutor’s i) use of expressions related to the task, and the ii) follow-up actions of such expressions. There is a crucial relationship between the two in this task, as interlocutors could simply be stuck saying. For this purpose, we investigate in RQ1 what interlocutors “say” at various phases in the dialogue (in H1.1), and how such expressions are surrounded by hesitation phenomena, i.e. the use of fillers like ‘um’ and ‘uh’ (H1.2). Then, RQ2 focuses on the relationship between expression and follow-up actions, at various phases in the dialogue (H2.1), and how the follow-ups are surrounded by information management phenomena, i.e. the use of ‘oh’. We also contribute a dataset of anonymised, transcribed children dialogues, event logs of their task progression, and code, which are all made publicly available to either reproduce our results or to use for further research. Our work highlights the need to bridge the gap between the fields of collaborative learning and dialogue analysis, and expand existing automatic measures for verbal behaviours, as they can have an impact on our understanding of the way children learn.

Considering the nature of the task, i.e. the situated environment, we limited our analysis to referents that were specific to the task. From a methodological perspective, this was done for several reasons: aside from being the minimal terminology interlocutors are required to use to ensure task success, it gives us insights into the way interlocutors (children) use (potentially unfamiliar) terminology related to the environment. Additionally, the referents are easy to automatically annotate from the transcripts without hand crafted annotations, and allow us to pinpoint specific areas of the activity where immediate effects on the situated environment are apparent – by observing the process of instructions evolving into actions. Looking at the results of our automatic methods, the benefits of this methodological choice are more apparent (for e.g. in Table 6), as we can see that the annotations can be thought of as periods of the task where there are immediate, situated actions. These periods, by inspecting the annotations, are about specific objects in the situated environment and can be thought of as “local contexts” in the timeline of the dialogue, i.e. the explicit reference to a node name; which results in actions that can be matched, mismatched or nonmatched. We complement the explicit use of these expressions with implicit markers of hesitation and information management. We thus decompose the overall use of these expressions and follow-up with actions into two aspects, i.e. ‘what was said and done’ and ‘how it was said and done’.

A first finding of this work is the discovery that the measures we propose are capable of capturing elements of lexical alignment in such a context. In terms of results, for RQ1, we see that all teams establish routines, regardless of task success. An assumption that might be commonly made, is that the more aligned interlocutors are, the more are chances of their task success (usually, measured by performance alone). Indeed, our results from RQ1 indicate that this is not necessarily the case for lexical alignment: we rather observed that better performing teams were earlier than badly performing teams to align by our measures. In terms of the formation of a routine, we see that hesitation phenomena tend to occur around establishment times, and greatly after priming times, indicating that the IF could be using fillers in the role of clarification. For RQ2, we observe that all
teams, regardless of task success, had matches and mismatches of instructions-to-actions. Similar to RQ1, we observed a general trend that better performing teams tend to follow up their instructions with actions earlier in the task rather than later. We see that information management phenomena tend to occur around the vicinity (specifically earlier) of (mis)matched actions, but the distributions overlap more for better-performing teams.

While our measures discussed in RQ1 and RQ2 do not show significant results for learning, we still think considering learning with performance is an important first step in evaluating task success in these activities, as performance does not necessarily bring about learning (Nasir, Norman, et al., 2020b). As an example we looked at Team 20, that performed well but did not learn. For this team, in H2.2, we saw that our measures suggest that the interlocutors were working in isolation, as evidenced by nonmatched actions. Our measures still reflect some fine-grained aspects of learning in the dialogue (such as an exploratory and collaborative period the interlocutors go through during the alignment process), even if we cannot conclude that overall they are linked to the final measure of learning. Our measures capture more aspects of performance than learning, as our measures focus on what was said specifically about the environment, and the immediate apparent changes to the environment – essentially the crux of the task. At a higher level, lack of understanding/learning etc. could be reflected by other phenomena (such as other multimodal features [Nasir et al., 2020a, 2021]), other expressions (“I don’t understand”) etc. Since there are many ways to learn, and hence different behaviours that could result in learning, it is unsurprising that such patterns are difficult to capture.

Working with dialogues is complicated, due to i) the lack of automatic evaluation criteria (e.g. human annotators might instinctively be able to say that a particular team collaborated well after observing a dialogue, but it is hard to empirically pinpoint the exact reasons such a judgement was taken), and ii) the nature of spontaneous speech (variable turn taking, disfluencies, etc.). Our results, albeit limited to a small dataset, highlight that in a situated activity, focusing simply on task-specific expressions and certain surrounding spontaneous speech phenomena can still give good insights into the nuances of collaboration between the interlocutors, and its ultimate links to task success (with the awareness that there are several other aspects that remain to be observed in the dialogue such as studying the gaze patterns, prosodic features and so on). However, work in collaborative learning tends to focus more on these non-verbal features; especially in automatic analysis. Future work will concentrate on other levels of alignment, for example prosodic cues. It would be interesting to look at the inferred instructions, rather than simply the (mis)match of these to actions, to see if this could give further insights into periods of collaboration and eventually learning amongst the interlocutors. We hope that the above findings can inspire further research on the topic and contribute to the design of technologies for the support of learning.

6. Acknowledgments

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 765955 (ANIMATAS Project).

7. Data Availability

All relevant data (transcripts, logs, and responses to the pre-test and the post-test, as well as the description of the network in the activity) are available from the Zenodo Repository, DOI: 10.5281/zen
The code that reproduces all the results and figures given in this paper are available from the Zenodo Repository, DOI: 10.5281/zenodo.4675070.

8. Declaration of Interest

There is no conflict of interest.

References


33


Jeanine Romano, Jeffrey D Kromrey, Jesse Coraggio, and Jeff Skowronek. Appropriate statistics for ordinal level data: Should we really be using t-test and cohen’s d for evaluating group differences on the nsse and other surveys? In annual meeting of the Florida Association of Institutional Research, volume 177, 2006.


Appendix A. Choosing a Representative Subset of the Dataset

We chose the subset to transcribe based on the following considerations:

- The percentage of successful teams (30% compared to 21% of the whole dataset).
- The distribution of teams in performance and learning outcomes. Fig. 3 shows how the teams are distributed in terms of performance (minimum error) and learning (relative learning gain). As the figure shows, the means of performance and learning of the transcribed subset are similar to those of the whole dataset.
- The distribution of the number of attempts (i.e. submissions) and turns. Fig. 9 shows how the teams are distributed in terms of the number of attempts and turns. The mean number of attempts and turns of the transcribed subset is similar to that of the whole dataset.

![Figure 9](image)

Figure 9: The distribution of transcribed tasks (red bars) and non-transcribed tasks (blue bars) in terms of the number of attempts (i.e. submissions) and number of turns. The mean of a set is shown as a vertical dashed line.

Appendix B. Pre-test and Post-test Details

The pre-test and the post-test are made up of 10 multiple-choice items with a single correct answer. The items of the pre-test and the post-test are defined in a context other than the context of the task (Swiss gold mines), and are based on variants of the graphics in the muddy city problem. All items were validated by experts of the domain and experts in education. The score of an item is 1 if it is answered correctly, and 0 otherwise. The pre-test and post-test assess the following concepts:

[33] https://csunplugged.org/minimal-spanning-trees/
Algorithm 1: \textsc{Recognise-Instructions} finds the edit instructions in an utterance. It is implemented in a script (\texttt{tools/\_recognise\_instructions\_detect\_follow-up\_actions.ipynb} in the tools) that generates an annotated corpus (\texttt{processed\_data/annotated\_corpus} available with the tools), where the column \textit{instructions} gives the list of instructions that are inferred for that row.

**Input:** A sequence of tokens \( U = \langle t_1, t_2, \ldots, t_n \rangle \) that make up an utterance \( U \)

**Output:** A sequence of instructions \( I = \langle i_1, i_2, \ldots, i_k \rangle \)

1. \( E \leftarrow \textsc{Recognise-Entities}(U) \)
2. \( I \leftarrow \text{an empty sequence} \quad \text{// for inferred instructions} \)
3. \( i \leftarrow \text{a new instruction object} \)
4. \( i.verb \leftarrow \text{null} \); \( i.u \leftarrow \text{null} \); \( i.v \leftarrow \text{null} \) // \( u \) is the first and \( v \) is the second node by mention
5. \( \text{foreach entity } e \in E \text{ do} \)
6. \( \quad \text{if } e.\text{label} = \text{`Add'} \text{ or } e.\text{label} = \text{`Remove'} \text{ then} \)
7. \( \quad \quad \text{if } i.\text{verb} \neq \text{null} \text{ then} \quad \text{// already inferring an instruction} \)
8. \( \quad \quad \quad \text{if } i.u \neq \text{null} \text{ then} \quad \text{// save the partial instruction} \)
9. \( \quad \quad \quad \quad \text{insert } i \text{ into } I \)
10. \( \quad \quad \quad \quad i.u \leftarrow \text{null}; i.v \leftarrow \text{null} \quad \text{// clear node 1 and 2} \)
11. \( \quad \text{else if } i.u = \text{null} \text{ then} \quad \text{// that is, } e.\text{label} = \text{`Node'} \)
12. \( \quad \quad i.u \leftarrow e.\text{token} \)
13. \( \quad \text{else if } i.v = e.\text{token} \text{ then} \)
14. \( \quad \quad \text{if } i.u \neq e.\text{token} \text{ then} \quad \text{// if not repeating node name} \)
15. \( \quad \quad \quad i.u \leftarrow e.\text{token} \)
16. \( \quad \quad \quad \text{if } i.v = \text{null} \text{ then} \quad \text{// default to a verb if not detected} \)
17. \( \quad \quad \quad \quad \text{if } I.\text{length} = 0 \text{ then} \quad \text{// no previous instruction: default to `Add'} \)
18. \( \quad \quad \quad \quad \quad i.\text{verb} \leftarrow \text{`Add'} \)
19. \( \quad \quad \text{else} \quad \text{// default to previous instruction’s verb if exists} \)
20. \( \quad \quad \quad i.\text{verb} \leftarrow I[I.\text{length} - 1].\text{verb} \)
21. \( \quad \quad \quad \text{insert } i \text{ into } I \)
22. \( \quad \quad i \leftarrow \text{a new instruction object} \) \( i.\text{verb} \leftarrow \text{null}, i.u \leftarrow \text{null}, i.v \leftarrow \text{null} \)
23. \( \text{return } I \)

1. **Concept-1** (existence): If a spanning tree exists, i.e. the graph is connected.
2. **Concept-2** (spanningness): If the given subgraph spans the graph.
3. **Concept-3** (minimumness): If the given subgraph that spans the graph has a minimum cost.

There are 3, 3, and 4 items associated with the concepts, respectively.

Appendix C. Algorithms

C.1 Instruction Recognition in RQ2

\textsc{Recognise-Instructions} automatically infers a sequence of instructions for an input utterance via a simple rule-based algorithm, as described in Algorithm 1. Note that allows inference of partial instructions i.e. that contain one node name only. It uses \textsc{Recognise-Entities} in Algorithm 2 to find the edit instructions in an utterance in a simple way.
Algorithm 2: RECOGNISE-ENTITIES finds the edit entities in an utterance via a simple rule-based named entity recognition procedure. It is implemented in a script (tools/6_recognise_instructions_detect_follow-up_actions.ipynb in the tools) that generates an annotated corpus (processed_data/annotated_corpus available with the tools, for which we employ named-entity recognition (NER) feature of the Python library spaCy) that performs this entity recognition.

Input: A sequence of tokens \( U = \langle t_1, t_2, \ldots, t_n \rangle \) that make up an utterance \( U \)

Output: A sequence of entities \( E = \langle e_1, e_2, \ldots, e_m \rangle \)

1. \( N \leftarrow \langle \text{‘Montreux’}, \text{‘Bern’}, \ldots, \text{‘Basel’} \rangle \) \hspace{0.5cm} // all node names
2. \( A \leftarrow \langle \text{‘add’}, \text{‘remove’}, \text{‘build’}, \text{‘connect’}, \text{‘do’}, \text{‘go’}, \text{‘put’} \rangle \) \hspace{0.5cm} // add verbs
3. \( R \leftarrow \langle \text{‘away’}, \text{‘cut’}, \text{‘delete’}, \text{‘erase’}, \text{‘remove’}, \text{‘rub’} \rangle \) \hspace{0.5cm} // remove verbs
4. \( E \leftarrow \) an empty sequence \hspace{0.5cm} // for inferred entities
5. foreach token \( t \in U \) do
6. \hspace{0.5cm} \( \text{label} \leftarrow \text{null} \)
7. \hspace{1cm} if \( t \in N \) then \( \text{label} \leftarrow \text{‘Node’} \)
8. \hspace{1cm} else if \( t \in A \) then \( \text{label} \leftarrow \text{‘Add’} \)
9. \hspace{1cm} else if \( t \in R \) then \( \text{label} \leftarrow \text{‘Remove’} \)
10. \hspace{1cm} if \( \text{label} \neq \text{null} \) then
11. \hspace{2cm} \( e \leftarrow \) a new entity object ; \( e.\text{token} \leftarrow t ; e.\text{label} \leftarrow \text{label} \)
12. \hspace{2cm} insert \( e \) into \( E \)
13. return \( E \)

C.2 Detecting Follow-up Actions of the Instructions in RQ2

MATCH-INSTRUCTIONS-TO-ACTIONS pairs instructions with actions as matches or mismatches, for a verbal and physical actions list \( A \), as described in Algorithm 3.

Preprocessing (to obtain a verbal and physical actions list \( A \)): We combine the transcript and edit actions in a subject-verb-object(-turn-attempt) format.

- Each utterance in the transcript is added as an action with the verb ‘says’.
- Each edit action from the logs is added with the verb ‘adds’ or ‘removes’, according to whether it is an add action or a remove action, respectively.

Each action \( a \in A \) has fields:

- \( a.\text{subject} \in \{ \text{‘A’}, \text{‘B’} \} \), the two learners that are collaborating to solve the given problem
- \( a.\text{verb} \in \{ \text{‘says’}, \text{‘adds’}, \text{‘removes’} \} \), the edit actions and utterance action for matching instructions with edit actions
- \( a.\text{object} \in \{ \text{Utterances} \} \cup \{ (u, v) : (u, v) \in \text{Edges} \} \)
- \( a.\text{turn} \in \{ 1, 2, \ldots, n \} \) indicating the turn number of the period the action belongs to (where for utterances, the start time of the utterance belongs to). After every two edits, the turn number incremented by 1
- \( a.\text{attempt} \in \{ 1, 2, \ldots, m \} \) indicating the attempt number of the period the action belongs to (where for utterances, the start time of the utterance belongs to). After every submission, the attempt number is incremented by 1
Algorithm 3: Match-Instructions-to-Actions pairs a list of pending instructions with actions as matches or mismatches. It is implemented in a script (tools/6_recognise_instructions_detect_follow-up_actions.ipynb in the tools) that generates an annotated corpus (processed_data/annotated_corpus available with the tools, where the column matching gives the the result of the matching for that row).

**Input:** A sequence of verbal and physical actions \( A = (a_1, a_2, \ldots, a_k) \)

**Output:** A sequence of \( M = (m_1, m_2, \ldots, m_k) \) holding (mis)match info \( m_i \) for each \( a_i \)

1. \( P \leftarrow \) an empty sequence for pending instructions to be matched
2. \( M \leftarrow \) an empty sequence for match/mismatch for each action in \( A \)
3. \( attempt \leftarrow 1 \) // submission no for clearing the pending instructions list
4. \( turn \leftarrow 1 \) // turn no for clearing the pending instructions list

foreach action \( a \in A \) do

// clear pending instructions if a new turn (or attempt i.e. submission)
if \( a.turn = turn + 1 \) then

| clear \( P \) // remove all items in the sequence \( P \)
| \( turn \leftarrow a.turn \) // update the current episode (i.e. new turn)

else if \( a.attempt = attempt + 1 \) then

| clear \( P \) // remove all items in the sequence \( P \)
| \( attempt \leftarrow a.attempt \) // update the current episode (i.e. new attempt)

// for say action, recognise instructions and update the pending instructions list
if \( a.verb = \textquote{says} \) then

| \( I \leftarrow \text{RECOGNISE-INSTRUCTIONS}(a.object) \) // \( a.object \) is the utterance
| foreach instruction \( i \in I \) do
| \( i.agent \leftarrow a.subject \) // set the instructing agent
| insert \( i \) into \( P \) // update the pending instructions

// for do action, try to match with a pending instruction
else if \( a.verb = \textquote{does} \) then

| \( I' \leftarrow \{ i: i \in I \text{ and } i.agent \neq a.subject \} \) // filter for the other interlocutor’s instructions
| \( m \leftarrow \) a new matching object ; \( m.match \leftarrow \text{null} \)
| if \( I'.\text{length} > 0 \) then // there is an instruction that may (mis)match

| foreach instruction \( i \in I' \) do
| \( m.match \leftarrow \text{True}; m.instruction \leftarrow i; m.action \leftarrow a \)

if \( m.match = \text{null} \) then // no matches, hence a mismatch

| \( i \leftarrow I'[I'.\text{length} - 1] \) // get the last instruction by the other
| \( m.match \leftarrow \text{False}; m.instruction \leftarrow i; m.action \leftarrow a \)

// process the match (if matched or mismatched)

if \( m.match \neq \text{null} \) then // match: True or mismatch: False

| \( M[i] \leftarrow m \) // add match object to list of matches
| // remove matching instructions from pending instructions sequence

foreach instruction \( i \in P \) do

| if \( \text{CHECK-MATCH}(i, a) \) then remove \( i \) from \( P \)

return \( M \)
Algorithm 4: CHECK-MATCH checks if an instruction matches with the action. It allows partial matching for partially inferred instructions (i.e. only one of the node names is mentioned).

**Input:** An instruction $i$ and an action $a$

**Output:** True if the intended action in $i$ and action $a$ match, False otherwise

1. if $i$.action $\neq$ $a$.verb then
2.     return False
3.     $u \leftarrow a$.object.u // first node in the edited edge
4.     $v \leftarrow a$.object.v // second node in the edited edge, sorted
5.     if $i$.v $\neq$ null then // instruction is partially inferred i.e. contains $i$.u only
6.         if $i$.u $=$ $u$ or $i$.u $=$ $v$ then // if one node matches
7.             return True
8.         else
9.             return False
10.     else if ($i$.u $=$ $u$ or $i$.u $=$ $v$) and ($i$.v $=$ $u$ or $i$.v $=$ $v$) then // both match
11.         return True // note that $i$.v $\neq$ $i$.u by its way of inference
12.     else
13.         return False