Discourse Structure for the Minecraft Dialogue Corpus

Kate Thompson[*†], **Julie Hunter**[*], **Nicholas Asher**[†‡]
[*]LINAGORA Labs, [†]IRIT, [‡]CNRS
Toulouse, France
{cthompson, jhunter}@linagora.com, asher@irit.fr

Abstract

We provide a new linguistic resource: The Minecraft Structured Dialogue Corpus (MSDC), a discourse annotated version of the Minecraft Dialogue Corpus (MDC; Narayan-Chen et al., 2019), with complete, situated discourse structures in the style of SDRT (Asher and Lascarides, 2003). Our structures feature both linguistic discourse moves and nonlinguistic actions. To show computational tractability, we train a discourse parser with a novel "2 pass architecture" on MSDC that gives excellent results on attachment prediction and relation labeling tasks especially long distance attachments.

Keywords: Corpus Creation, Discourse Annotation, Discourse Parsing, Situated Collaborative Tasks for HRI, Natural Language Instruction Following

1. Introduction

A human using situated conversation to guide a collaborating agent through a task typically appeals to a number of familiar strategies. One is to break the task down into a series of smaller ones, moving from one instruction to the next as the subtasks are successfully completed. In order for this strategy to succeed, the human generally needs to elaborate on instructions, to monitor and to acknowledge the agent's performance and to correct the agent whenever failure occurs. To facilitate the task, the agent should be able to ask clarifying questions to which the human can respond in order to avoid mistakes.

We need models for understanding such linguistic interactions if we are to move from manually programming automated agents like cobots to instructing and teaching them through natural conversation that exploits nonlinguistic contextual information. The Minecraft Dialogue Corpus (MDC; Narayan-Chen et al., 2019) provides the opportunity to study language use in the context of such conversations. In the task, a Builder receives instructions from an Architect and then attempts to execute the instructions in a simple Minecraft world. Jayannavar et al. (2020) exploited this corpus to train a deep learning model (Neural Builder) that executes Minecraft building actions given natural language input. However, the results on this task show that inferring excecutable actions from language in the Minecraft setting is difficult; the best F1 scores from Jayannavar et al. (2020) that we have independently verified are close to 0.2.

The task is difficult in part because successfully communicating a single instruction in the corpus—as in real life—often requires complex conversational interactions that develop over several turns. The Architect may elaborate on previous utterances to clarify missing details or correct previous action or instruction sequences, while the Builder may ask questions. In addition, conversational interactions in the MDC heavily exploit the nonlinguistic environment in which a conversational exchange takes place, and discourse moves like answering a question are sometimes realized not in words but with nonlinguistic actions.

We believe that if the Builder can effectively compute the right conversational structure from such linguistic exchanges and information about the nonlinguistic context, he or she will be more likely to succeed in the building task. This paper takes a first step towards testing this hypothesis by making two contributions:

1. We release, as a resource for the NLP and robotics communities, the Minecraft Structured Dialogue Corpus (MSDC): [1] a set of full discourse structures for all MDC dialogues in the style of Segmented Discourse Representation Theory (SDRT; Asher, 1993; Asher and Lascarides, 2003; Asher et al., 2016).

2. We describe and evaluate a simple but innovative two stage discourse parser that effectively learns MSDC discourse structures including long distance relations.

The outline of our paper is as follows. In Section 2, we present background on the MDC and discuss certain features that make it particularly challenging. In Section 3 we present the MSDC and show how it addresses some of those challenges. In Section 4, we describe some important patterns that we have found in the annotations. Section 5 details our parsing experiments on the MSDC. We conclude in Section 6.

[1] https://github.com/linagora-labs/MinecraftStucturedDialogueCorpus

2. The Minecraft Dialogue Corpus

The corpus consists of 547[2] dialogues in which two humans, in the roles of Architect and Builder, collaborate to construct a three-dimensional object in a simulated Minecraft world. The Builder has blocks in 6 different colors that can be placed inside an 11 x 9 x 11 grid. The Architect cannot build, but guides the Builder by typing instructions in a chat window. The Builder can ask questions and comment on what they have done or on the Architect's instructions. The dialogue finishes when the Architect judges that the Builder has successfully completed the construction.

The MDC involves incomplete information, as only the Architect has knowledge of the structure to be built. This gives rise to linguistic features that are emblematic of many if not most cooperative tasks that we imagine for human-cobot interactions. First, arriving at a common understanding of instructions or of which objects and actions are under discussion at a particular point requires a conversational negotiation between Builder and Architect. Second, to build a complex object, the Architect must break the process down into simpler steps—a feature which helps us understand compositionality in a new way. Studying this corpus and its discourse structure thus carries lessons far beyond Minecraft data.

The MDC is a situated corpus, containing not only chat moves but also pick and place moves by the Builder that are perceived by both players and for which descriptions can be recovered from game logs. In the text-based version of the corpus, moves are described in terms of x, y, z coordinates; e.g., `Builder puts down a blue block at X:3, Y:5, Z:0`. The situated nature of the corpus leads to discourse relations between non-linguistic and linguistic moves, as when players comment on, question and correct pick and place moves. It also leads to rich and varied spatial language, which has provided the basis of a separate but complementary annotation campaign (Bonn et al., 2020) extending the framework of Abstract Meaning Representation (Banarescu et al., 2013).

3. The Discourse Annotations

The MSDC provides full discourse structures in the style of SDRT to dialogues in the MDC. Other theories of discourse structure have been proposed, such as RST (Mann and Thompson, 1987), LDM (Polanyi et al., 2004), the Graphbank model (Wolf and Gibson, 2005), DLTAG (Forbes et al., 2003), and PDTB (Prasad et al., 2008). We opt for SDRT because it, unlike other frameworks that impose tree structures as full discourse structures, only requires that discourse structures be represented as acyclic, directed graphs with a unique head. This allows for a single discourse unit to have more than one incoming link, or *multiple parents*, in the graph. Multi-parent units are a common occurrence in the MSDC. Figure 1 illustrates multiple examples, including one in which the Builder places a blue and then a purple block and asks the Architect: "like that?". The Architect responds, "not quite," which not only serves to answer the Builder's question but simultaneously to correct the Builder's last moves. The full content of the Correction is specified through the following two elaborating discourse moves. Using SDRT also facilitates comparison with the STAC corpus (Asher et al., 2016), the only large-scale, multiparty situated dialogue corpus that has been annotated for discourse structure.

The dialogues in the MDC range in length from 10 to over 200 turns. To build a discourse graph, we decompose each turn in a given dialogue into what are called *elementary discourse units* (EDUs). EDUs are the basic building blocks of an SDRT discourse structure and correspond roughly to clauses. Because the MDC conversations are situated, our discourse graphs also contain *elementary event units* (EEUs), the basic Builder actions described in Section 2.

We also include, in a limited way, *complex discourse units* or CDUs. A CDU is a group of discourse units that work together to provide a single argument to a discourse relation. For example, if someone says, "I had a bad day. I lost my phone and I broke my ankle", the second sentence can be broken down into two EDUs which *together* provide an explanation for the bad day. To simplify downstream discourse parsing, we decided to avoid CDUs between linguistic units. We did, however, include them for action moves. Most actions in the corpus take the form of a series of consecutive pick and place moves. The relations between these moves are very regular: as one move simply happens after another, we would normally relate each pair of consecutive action moves via Sequence. However, since the large number of such links would complicate discourse parsing, and "drown out" the data on more significant but more complicated relations in our corpus, we decided to "squish" each sequence of actions contained in a CDU into one long EEU by concatenating its parts.

Each EDU or EEU bears a *discourse relation* to one or more other discourse units. To the usual SDRT relations (apart from Background and Parallel), we added the relation type Confirmation Question, because Builders frequently ask Architects to confirm whether an action they have just

[2] The original corpus is reported to have 509 dialogues, but this does not include the practice interactions also included in the dataset.

	Train+Val	Test	Total
Original MDC			
# Dialogues	410	137	547
MSDC			
# Dialogues	407	134	541
# EDUs	17135	5417	22552
# EEUs	25555	7263	32818
# EEUs *squished*	4687	1475	6162
# Relation instances	26299	8275	34574
# MP DUs	4798	1482	6280
# Speaker turns per dialogue:			
Min	6	9	6
Max	105	69	105
Mean	31.6	29.5	31.1
# DUs per dialogue:			
Min	8	11	8
Max	186	120	186
Mean	53.6	51.4	53.1

Table 1: MDC and MSDC characteristics. We did not annotate 6 problematic games in the MDC.

performed is correct (e.g., “‘like that?” in Figure 1). All 16 relation types are listed in Table 2.

Three linguists and two NLP experts annotated all the dialogues of the MDC using the GLOZZ annotation tool (Mathet and Widlöcher, 2009). Two different linguists have reviewed each annotation twice. We also employed scripts to check for various annotation errors.

Table 1 shows statistics about our corpus, breaking down counts according to the original train-test split used in (Jayannavar et al., 2020), with 137 dialogues for testing and the 410 for training and development. The MSDC provides annotations for the majority of the dialogues in the original MDC, in which the speaker turns are broken down into EDUs and squished EEUs.

Even with squished EEUs, the MSDC is larger than the purely linguistic/non-situated version of the STAC corpus and it is a little more than half the size of the situated STAC corpus (Asher et al., 2020). With unsquished EEUs, the MDC is about the size of the STAC corpus.

4. Discourse Structural Characteristics of MDC

The MSDC reveals several typical and important features of discourse structure for instructional dialogues involving a cooperative task where agents interact with the nonlinguistic environment.

4.1. Negotiation Sub-dialogues and Narrative Arcs

In Figure 1, the Architect gives an instruction (“place a blue block one block to the right ...”) that the Builder does not completely understand. The Builder could have asked the Architect directly to clarify or elaborate on the instruction, but instead opts for a strategy of first building to illustrate their understanding and then asking for confirmation. The Architect responds by correcting the Builder’s actions, and the latter then corrects the original action sequence. Finally, the Architect acknowledges the acceptability of the current state of construction state and moves to the next instruction.

This kind of conversational negotiation between Builder and Architect, which allows them to arrive at a common understanding of what actions are to be executed, is common in the MDC and gives rise to a very regular, high level structure consisting of a sequence of *negotiation episodes* connected by Narration instances (Narrations). Each episode begins with a new instruction, continues through a stage in which the Architect may give additional instructions or feedback and the Builder may ask questions, and concludes when the final construction is complete or when the Architect moves to a new instruction, usually after acknowledging the completion of the last instruction. Figure 1 shows one such episode.

The narrative arcs linking negotiation episodes give the structural backbone and macrostructure of the conversation (Asher et al., 2020). This macrostructure gives an overview of how the two interlocutors went about the construction task and is an important feature for human understanding of how complex tasks are decomposed into simpler ones. Table 2, which gives the distribution of distances in terms of DUs between arguments of all relation types, shows that Narration relation instances often cover longer distances. In fact, Table 2 shows that Narration supports more long distance relations in the MSDC than any other relation. The MSDC differs markedly from conversational corpora like STAC, where Narrations were scarce and relatively short distance (Asher et al., 2016, 2020).

Linguistic corrections by the Architect, like that shown in Figure 1, are a particularly important structure during the negotiation episodes. In such cases, the Architect generally corrects a Builder action a with a linguistic move ℓ—by which we mean that ℓ contains an instruction that gives information on how to revise or modify a. This linguistic correction ℓ then results in a nonlinguistic action b by Builder that itself serves as a *nonlinguistic* correction of a. By a nonlinguistic correc-

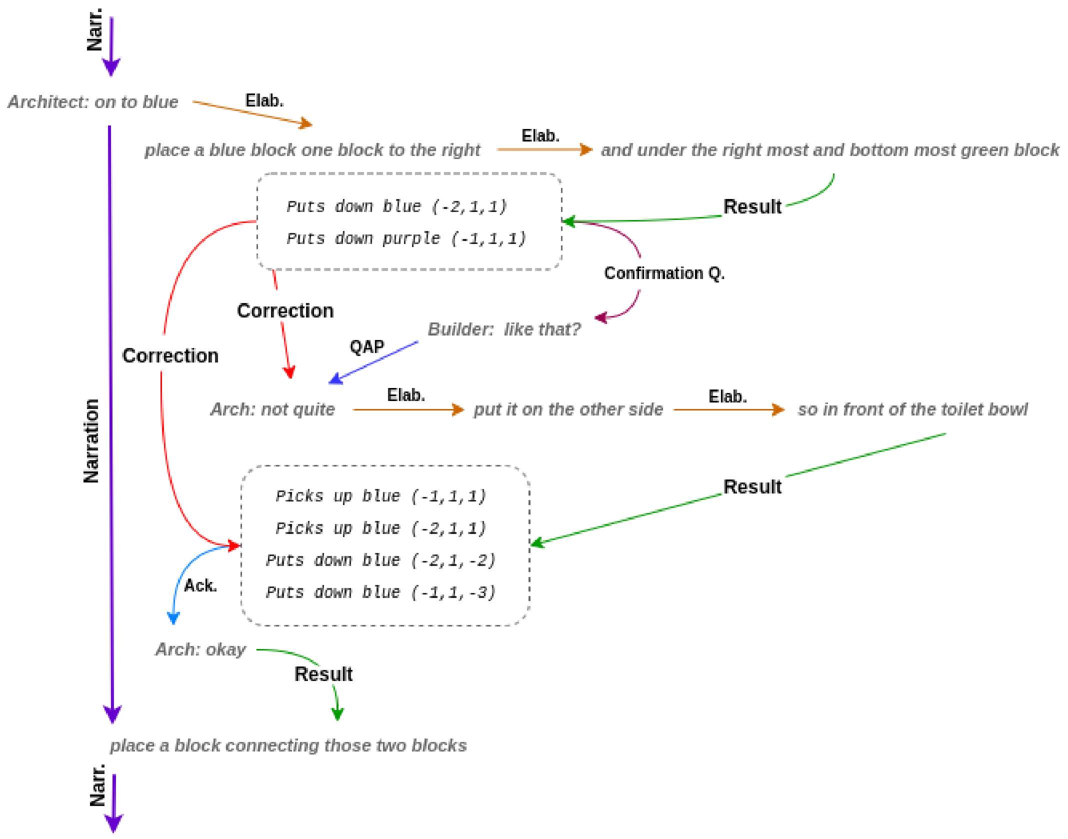

Figure 1: MSDC example showing CDUs composed of EEUs for action sequences, non-treelike structures and multi-parent EDUs. The Narrations on the left give the narrative arc/macrostructure of the dialogue, while the structure between the arguments of the Narrations shows a negotiation episode with Correction.

tion, we mean that the action b undoes something in or somehow revises the action a. The presence of multiple Architect corrections are to be expected in a situation of asymmetric information and an imperfect communication channel, two features which are typical for instructional dialogues.

4.2. Multi-Parent Structures and Nonlinguistic Arguments

As noted in Section 3, the Architect's reply "not quite" in Figure 1 simultaneously answers the Builder's confirmation question and corrects their previous actions.[3] The MSDC features many cases in which a single EDU has multiple incoming relation instances, and thus multiple parents, resulting in non-treelike structures.

The second argument of a narrative arc spanning a negotiation episode is always a multi-parent EDU, as it is a new instruction that both sequentially follows the preceding instruction and also *results* from the success of the previous linguistic-action sequences. In Figure 1, the instruction "place a block connecting those two blocks" is an example. New instructions can result from an acknowledgment of building success (Figure 1), a comment (e.g., "great!"), a positive answer to a confirmation question or directly from a successful action sequence/EEU. Table 3 shows that Narration and Result instances have some of the highest occurrences in multi-parent structures.

Table 1 gives the total number of multi-parent units ("MP DUs"). The participation of the different relation types in multi-parent units is shown in 3. For instance, Result and Narration have the highest numbers of relations connecting MP DUs, while Narration, and Correction have a high proportion of their total relations connecting MP DUs.

In the STAC corpus, non-treelike structures are often attributable to complex interactions between nonlinguistic actions and linguistic moves. The MDC has similar interactions but in much greater numbers. STAC conversations are directed to-

[3] Ideally, we would group the EDU expressed by "not quite" in a CDU along with the following two elaborating moves, as they serve together to correct the action sequence and answer the Builder's question. As explained in Section 3, however, we made the simplifying choice to not use CDUs for linguistic moves.

	1	2	3	4	5	6	7	8	9	10	>10	Max len	Total
Result	8847	1091	295	115	43	25	8	6	1	1	1	12	10433
Elaboration	3627	348	69	16	9	10	3	2	0	0	2	13	4086
Narration	155	983	878	577	419	334	263	161	142	124	425	63	4461
Acknowl.	3529	820	134	38	14	11	4	2	1	0	1	13	4554
Correction	528	564	460	295	143	78	41	23	7	9	12	20	2160
Q-A Pair	1524	290	84	26	8	3	1	0	0	0	0	7	1936
Comment	1437	181	50	10	2	0	0	0	1	0	0	9	1681
Continuation	1367	318	176	95	38	23	7	2	4	1	0	10	2031
Confirmation-Q	885	85	22	6	1	0	0	0	0	0	0	5	999
Clarification-Q	612	256	49	19	12	3	3	1	1	2	2	14	960
Q-Elab	151	62	14	3	0	0	0	0	0	0	0	4	230
Contrast	368	17	8	5	1	0	0	0	0	0	0	5	399
Sequence	19	14	2	1	1	1	0	0	0	0	0	6	38
Explanation	98	8	1	1	0	0	0	0	0	0	0	4	108
Alternation	162	9	1	0	0	1	0	0	0	0	0	6	173
Conditional	58	5	2	2	0	0	0	0	0	0	0	4	67
Backwards													
Comment	238	4	0	0	0	0	0	0	0	0	0	2	242
Conditional	16	0	0	0	0	0	0	0	0	0	0	1	16

Table 2: Relation type counts by distance

wards trade negotiations and actions; the MDC features collaborative multimodal tasks involving a wider variety of actions with both linguistic and nonlinguistic effects. STAC, like MOLWENI (Li et al., 2020), also includes multiparent structures that are attributable to the multi-party dialogue set up, which is absent in the MSDC. The MSDC shows that instructional situated dialogue typically gives rise to a rich and novel variety of non-treelike structures, many of which can be attributed to the difficulty of providing clear and unambiguous instructions in such a task.

The non-treelike structures described above exhibit complex interactions between linguistic and nonlinguistic contexts. Table 3 shows how many relations in the MSDC have nonlinguistic and linguistic arguments. Especially noteworthy are the very frequent Correction relation instances where one action sequence corrects the effects of another action sequence. This feature seems inevitable in a situation where two agents with asymmetric information are trying together to do some cooperative task, but it is not present in other multimodal dialogue corpora as far as we know.

5. Discourse Parsing with the MSDC

To establish a baseline for structure prediction for the MSDC, we used the BERTLine parser (Bennis et al., 2023). Its simple architecture gives robust results on multi-party, situated conversation data from the STAC corpus (Asher et al., 2016), which makes it a suitable starting point for the MSDC.

We followed the test split from Jayannavar et al. (2020) of our 541 annotated dialogues, except that we took 32 as a hold-out set for development of the parser.

5.1. The Parser

As is standard procedure in discourse parsing, we first segment the data into DUs. In our case, only the EDUs were segmented, since the EEUs were recorded as distinct moves (Section 2). The next step is to decide, for each pair of DUs, if they are attached. Finally, a relation type label is predicted for each positive attachment.

BERTLine takes EDU pairs as inputs. We did not consider every possible candidate pair for a segmented dialogue, however, as the number of candidates per dialogue is significantly greater than the number of relation instances in each dialogue, leading to significant class imbalance. We opted instead to focus on candidates that are of a distance that that captures the large majority of relation instances in our corpus. Table 2 shows that many of the relation instances have distance less than 7—especially those occurring most frequently within negotiation episodes, e.g. Result, Elaboration, Acknowledgement, Confirmation Question and Question-answer Pair. However, as we explain below, we ultimately took our final cutoff distance to be 10, in order to capture more of the long distance Narrations crucial for delineating negotiation episodes. To further mitigate the imbalance, we undersampled the training set so that the unattached/attached candidate ratio is no more than 3:1. Our corpus also has a number of

	Ling-Ling	Ling-NL	NL-Ling	NL-NL	Multi-parent
Result	1947	6109	2377	0	5234
Elaboration	4086	0	0	0	128
Narration	4461	0	0	0	4283
Acknowledgement	1816	0	2738	0	474
Correction	230	0	963	967	1477
Question Answer Pair	1936	0	0	0	797
Comment	1759	0	164	0	27
Continuation	2031	0	0	0	115
Confirmation Question	43	0	956	0	15
Clarification Question	960	0	0	0	5
Question Elaboration	230	0	0	0	6
Contrast	399	0	0	0	110
Sequence	0	38	0	0	0
Explanation	108	0	0	0	2
Alternation	173	0	0	0	11
Conditional	83	0	0	0	17

Table 3: Relation type counts by argument types, including backwards relation counts. For backwards relations types, see Table 2.

	Precision	Recall	F1
Trained on d10			
Finetuned	0.8584	0.7375	0.7933
+Linear (L10)	0.8278	**0.7507**	0.7874

Table 4: Attachment scores for finetuned BERT and finetuned BERT plus linear layer (BERTLine), both trained and evaluated on candidate sets of max distance 10.

relation instances where the order of arguments is reversed (backwards relations in Table 2), but we ignored these in the parsing.

We used the selected candidate pairs and relation instances to finetune a BERT model (Devlin et al., 2018) on binary attachment. Since this model relies on BERT embeddings, we needed to further preprocess the EEUs into special BERT tokens, with a unique token for each possible block color and placement combination. Single moves are represented by a single token, while EEU sequences (Section 3) are sequences of tokens.

We fine-tuned on attachments for candidates at distances $\leq$ 7 and $\leq$ 10 and tested on attachments of distance $\leq$ 10. The finetuned distance $\leq$ 7 model was about par with distance $\leq$ 10 models on precision but did less well than the latter on recall, and performed particularly poorly on longer distance relations. To improve recall on these crucial relations, we chose a distance of 10. We also added a simple linear layer to incorporate speaker change and distance information. The addition of this layer boosted the parser's recall on the MSDC at the cost of some precision. Table 4 shows the results for our fine-tuned BERT and the final BERTLine model with linear layer (L10).

We took the attachment predictions of the L10 model as input to the last component of the parser, which predicts a relation type label for each predicted (positive) attachment. This uses a multitask architecture to capture the informational dependencies between attachment and labeling decisions, and is described in detail in Bennis et al. (2023). Table 5 shows F1 scores for each relation type and overall average F1 on two different sets of attachments: the full predicted attachments, and the subset of those attachments that were correctly predicted (true positives).

Our labeling score was decent given our attachment score, which is high for a dependency style discourse parsing task (Morey et al., 2018). The scores for labeling relative to both the predicted true attachments and all predicted attachments can be found in Table 5, which shows that our parser is competitive with other parser performance on similar SDRT-annotated dialogue corpora (Shi and Huang, 2019; Wang et al., 2021; Bennis et al., 2023).

While recall is improved by including longer distance relations in training, BERTLine's performance on attachment and relation labeling—like that of many other discourse parsers—still degrades significantly on these relations. This is to be expected, as the ratio of attachments to candidate pairs drops dramatically as relation distance increases. Narration and Correction are two relation types in MSDC that most frequently occur at longer distances (see Table 2); and their F1 suffered as we expected (see Table 5). Nevertheless, these two relation types are indispensable to understanding not just the content of instructions, but

	F1: attach	F1: pred-TP	Δ
Result	0.85	0.98	13
Elaboration	0.75	0.84	9
Narration	0.50	0.93	43
Acknowl.	0.81	0.94	13
Correction	0.31	0.76	45
Q-A Pair	0.76	0.96	20
Comment	0.50	0.66	16
Contin.	0.44	0.56	12
Confirm-Q	0.86	0.98	12
Clarif-Q	0.61	0.88	27
Q-Elab	0.38	0.67	29
Contrast	0.80	0.88	8
Sequence	0.0	0.0	-
Explan.	0.0	0.0	-
Altern.	0.88	0.91	3
Conditional	0.58	0.67	9
Macro F1	0.53	0.73	
Weighted F1	0.60	0.89	

Table 5: F1 scores for L10 predictions on each relation type (listed in descending order of relation frequency in Table 2) on *all* predicted attachments, on *all correct* predicted attachments (TP only), and the percentage point gain in F1 when only TP attachments are considered.

the high level structure and break down of the task. In the rest of this section, we discuss one possibility for improving performance of our current model architecture at longer distances, and outline a preliminary experiment.

5.2. The Second Pass

The semantic connections that discourse structures represent leverage dynamic *global* information, including nonlinguistic context and previous discourse moves. Discourse parsers that consider a single candidate pair at a time can only make *local* attachment and labeling decisions, despite attempts to remedy the deficiency (Shi and Huang, 2019). While BERTLine might reliably predict much of the local structure for a situated exchange like the ones in the MSDC, it will inevitably fail to predict attachments and relation types whose semantics are not really learnable from the local, linguistic information of the candidate DUs.

BERTLine's linear layer makes a first attempt to add global cues to each local decision, but as observed in the MSDC, correctly characterizing the discursive import of certain speaker moves requires a more sophisticated understanding both of the instruction state and the task state. We see this with Narration and Correction attachment decisions: while Narration requires understanding when a subtask has been completed, Correction requires interlocutors to correctly identify the action and the portion of the world state that need to be changed, and in what way. These are hard problems that BERTLine cannot fix on its own.

We hypothesize, however, that some relations with longer distance instances *supervene* upon structures learned inductively from more local features. That is, some instances of longer distance relations might be inferrable from local discourse configurations. This turns out to be the case with long distance instances of Narration in the MSDC, and, we further conjecture, in dialogue involving asymmetric, collaborative tasks like that in the MDC more generally.

In the case of Narration, we know that longer distance Narration instances that connect subtasks in an MSDC dialogue link Architect EDUs. These architect EDUS are the second argument of a Result relation instance whose first argument is usually the last, previous completed Builder move or the Architect's acknowledgement of that move. These Result relation instances are quite local (distance of 1 or 2 EDUs between the arguments). Given perfect information about Result and the type and speaker of a given discourse unit, we should be able to algorithmically predict the longer distance Narration instances, thus establishing the supervenience of the latter upon the former.

To capture these longer distance Narration instances that form the Narrative arc of the session, we ran a second model that could exploit the outputs of the first. Having looked at the Narrative arcs on the development set, we saw that instead of using a second instance of BERTLine, we could use a deterministic algorithm to exploit information from BERTLine's first pass.

Recalling that a full MSDC session G is a list of EDUs, we can retrieve, after a first pass on G by BERTLine for each EDU, information about its speaker and its role in the discourse structure. In particular, we can determine a first EDU of the first Narration instance, since that is linked by Continuation from the first turn of G. And for any subsequent EDU, we can determine whether it is the second argument or *target* of a Result relation instance whose first argument or *source* is the previous Builder move or an architect's move that acknowledges the previous Builder move. Below let R stand for *Result*, C for *Continuation* Ak for *Acknowledgment*, and let $AR(e_1, e_2)$ stand for the fact that e_2 is the first architect move after edu e_1 and $BM(e_1, e_2)$ for the fact that e_1 is the last builder move before architect move edu e_2. Let *Feat*(e_i, e_j), for EDUs e_i, e_j stand for features given by $1stpass$, the BERTLine first pass. These include information about sources and targets of relations, and also types of moves.

The idea of the algorithm is that we go through each EDU sequentially, keeping track of whether

Algorithm 1 Algorithm for long distance Narration instances in MDC dialogue G

```
For $e_i \in (e_1, ...e_n)$ :⇐ EDUs of G
Feat$(e_j, e_i), 1 < i \leq n$ :⇐ 1st pass
if $C(e_0, e_i) \in$ Feat$(e_0, e_i)$ and $AR(e_0, e_i)$ then
$Narr(e_0, e_i)$
else
  if $e_k > e_i$ then
    while $k < n$ do
      if [[R$(e_j, e_k) \in$ Feat$(e_j, e_k)$ and last-BM$(e_j, e_k)$] or [last-BM$(e_j, e_k)$ and Ak$(e_j, e_l) \in$ Feat$(e_j, e_l)$ and R$(e_l, e_k) \in$ Feat$(e_l, e_k)$]] then
      Narr$(e_i, e_k)$ and $e_k \leftarrow e_{k+1}$ and $e_k \leftarrow e_i$
      else $e_k \leftarrow e_{k+1}$
      end if
    end while
  end if
```

an action has occurred. When we find ourselves in the state in which an architect's turn is linked by Result to a previous action or acknowledgment of an action, we consider this the end of the current narrative arc and the beginning of the new one.

As can be seen from Table 6, even with just BERTLine's predictions on Result, Continuation and Acknowledgement, the algorithm provided consistently superior results over BERTLine for Narration instances of distance > 2. With gold input on Result and Continuation and Acknowledgment relations, the F1 score for the algorithm was near perfect, with an F1 of 0.96 on all Narration instances.

To optimize the integration of the second pass with BERTLine's first pass of discourse annotations, we needed to see where BERTLine could be useful for Narrations and where the algorithm took over and gave much better results. We analyzed the predictions of the second algorithm on our development set, along with BERTLine's predictions for Narration. We compared their performance with respect to links of different lengths to decide at what point we should turn over the Narrations to the algorithm and the second pass. Our results are in Table 6. There were very short and intra turn Narration links that our algorithm did not take account of. We chose a cutoff at Narration relation instances of length < 2 and intra turn Narration instances to be done by BERTLine with the rest contributed by the algorithm.

We then integrated the effects of the first and second pass together. Because we did not take certain labeled predictions from BERTLine's first pass (Narration instances of length > 2) and added new labeled predictions with the Second Pass, we had to revise our scores for attachment as well as relation labeling. Given that relation labeling assigns a unique label to each attachment pair, we recomputed the new attachment F1 by subtracting the unused predictions from the first pass and adding the attachments predicted by the second pass algorithm. To compute an overall relation labeling score, we took the aggregated Narration score and combined it with BERTLine's first pass predictions for the other relations on the labeling task. The results are in Table 7. We note that BERTLine's performance on the intra turn and short inter-turn Narration links was poor on the development set, with a total F1 of 0.16. This lowered the first + second pass performance on Narration overall.

Table 7 shows that the first + second pass improves upon BERTLine's or the first pass's attachment and relation labeling scores when the evaluation is restricted to candidates of distance $d \leq 10$. More importantly, when we look at its predictions for an entire MSDC dialogue, the first + second pass loses nothing in precision and very little in recall from its performance on candidates of $d \leq 10$. A similar story holds for relation labeling. Our new first + second pass architecture thus achieves a very good performance on long distance attachments and their labels, something that has eluded discourse parsing in the past.

Our first + second pass architecture extends in principle to other relations like Correction, which also has long distance instances in the MSDC. Many Correction instances occur between EEUs (see Table 3) with a certain discourse configuration, and an algorithmic approach should be able to capture these. We did not, however, have time to implement a similar, algorithmic Second Pass approach for Correction. Instead, we tried simply running a second pass with BERTLine for which we added more local discourse structure features to the model's linear layer. We failed to improve the parser's performance with this approach as opposed to an algorithmic one, and we see two possible explanations for this failure. First, BERT has no pretraining data on the encodings of the nonlinguistic actions, so it couldn't easily learn patterns involving them. To test this, we switched values in the EEU encodings that would entail a different relation than Correction, but BERTLine did not pick up on these differences. Secondly, our BERT encoding of EEUs is not optimal, which made the task even more difficult.

6. Related Work on the Minecraft Corpus

Bonn et al. (2020) annotated sentences in the MDC using the AMR formalism (Banarescu et al., 2013). Bonial et al. (2021) extended those annotations on the MDC with labels for dialogue acts in the AMR formalism (Bonial et al., 2020). These partially overlap with discourse annotations but do not capture the full structure of the interactions. Combining the annotations of Bonn et al. (2020)

	Dev.Gold	BERTLine		Algo.
	# Narr	# Narr predicted (F1)		
Dist.	*intra/inter*	*intra*	*inter*	*inter*
1	1/0	7 (0.0)	-	-
2	3/61	5 (0.50)	56 (0.85)	63 **(0.91)**
3	1/73	0	51 (0.88)	65 **(0.92)**
4	0/34	0	31 (0.72)	25 **(0.78)**
5	0/37	0	26 (0.55)	27 **(0.80)**
6	0/22	0	17 (0.55)	17 **(0.73)**
7	0/21	0	12 (0.53)	17 **(0.80)**
8	0/8	0	4 (0.33)	4 **(0.62)**
9	0/6	0	3 (0.00)	7 **(0.29)**
10	0/7	0	2 (0.22)	11 **(0.77)**
Overall F1:		0.50		**0.76**

Table 6: Narration First and Second Pass predictions and F1 scores by relation distance (Dist) on the development set. *Intra* refers to intra-turn links, and *inter* to inter-turn links.

	BERTLine	1st+2nd Pass	
	$d = 10$	$d = 10$	$d = \infty$
Attachment			
Precision	0.82	0.82	0.82
Recall	0.75	0.79	0.78
F1	0.78	0.80	0.79
Relations			
Narration F1	0.50	0.73	0.69
all Macro F1	0.53	0.55	0.54
all Weighted F1	0.60	0.62	0.61

Table 7: BERTLine's results (left column) on attachment and relation label prediction from tables 3 and 4 compared to the combined results for the first + second pass with a distance 10 and no distance cutoff.

and Bonial et al. (2021) with MSDC will provide a detailed linguistic view of the Minecraft corpus at several different levels.

With regards to the relevant discourse literature, there has been work on instructional dialogues since the earliest days of computational work on discourse (Deutsch, 1974; Grosz and Sidner, 1986). The first discourse annotated corpus on instructional texts, with a single author and no extra-linguistic information, in a current formalism (RST) we know of is described in Subba and Di Eugenio (2009), for which Subba and Di Eugenio developed an RST-inspired parser.

STAC (Asher et al., 2016, 2020) is an SDRT-style discourse annotated corpus of multi-party dialogues that consist of chat exchanges between three or four players while playing an online version of the *Settlers of Catan* board game. STAC involves annotations of both linguistic and nonlinguistic actions or states in the discourse structure and is thus a situated dialogue corpus that resembles the MDC. However, the STAC dialogues are not fully collaborative but rather strategic. STAC also contains linguistic CDUs. We tried these for the MSDC but we found they cluttered the annotation too much. Researchers in parsing have "flattened" CDUs from SDRT annotations to just graphs over EDUs and EEUs (Perret et al., 2016; Shi and Huang, 2019; Wang et al., 2021). Bennis et al. (2023) also proposes squishing for CDUs with EEUS. Molweni (Li et al., 2020) is another corpus of multi-party dialogues, involving questions with answers from several interlocutors, from an Ubuntu chat corpus annotated with SDRT-style discourse structures. It does not involve any non-linguistic information or context.

Shi et al. (2022) try to predict when one should execute an action and when they should instead ask for a clarification question using a version of the MDC edited with dialogue acts. They annotated all Builder dialogue moves with a taxonomy of dialogue acts and then specified a *single* specific action under the execution label. Thus, their set up is not directly comparable to that of Jayannavar et al. (2020). Their dialogue annotation is also quite different from ours as it only labels Builder moves.

7. Conclusion & Future Work

We have introduced the MSDC, a new discourse annotated corpus for the Minecraft dialogues collected by Jayannavar et al. (2020). We have shown that the annotations are computationally tractable with a simple discourse parser that exploits BERT EDU encodings. We've also explored a new way of thinking about building discourse structures in multiple passes that captures long distance relations of certain types far better than other discourse parsers. Information from a first pass by the discourse parser helped improve scores for certain relations with long distance instances.

In future work, we will exploit discourse structure to improve the accuracy of an automated Builder's actions in response to MSDC instructions.

8. Acknowledgements

For financial support, we thank the National Interdisciplinary Artificial Intelligence Institute ANITI (Artificial and Natural Intelligence Toulouse Institute), funded by the French 'Investing for the Future–PIA3' program under the Grant agreement ANR-19-PI3A-000, and we also thank the ANR project COCOBOTS (ANR-21-FAI2-0005) and and the ANR/DGA project DISCUTER (ANR-21-ASIA-0005). We also thank the COCOPIL

“Graine” project funded by the Région Occitanie of France.

8.1. Ethical considerations and limitations

Our new resource has no ethical impact of which we are aware. The limitations of our study is that it is only a first step in gauging the importance of discourse structural information in improving language to code tasks and human/cobot interactions on collaborative tasks using language and vision. At the same time, we hope that the corpus and its discourse annotations will be useful to the community and may serve other purposes as well.

9. Bibliographical References

Nicholas Asher. 1993. *Reference to abstract objects in discourse*, volume 50. Springer Science & Business Media.

Nicholas Asher, Julie Hunter, Mathieu Morey, Farah Benamara, and Stergos D Afantenos. 2016. Discourse structure and dialogue acts in multiparty dialogue: the STAC corpus. In *LREC*.

Nicholas Asher, Julie Hunter, and Kate Thompson. 2020. Modelling structures for situated discourse. *Dialogue & Discourse*, 11:89–121.

Nicholas Asher and Alex Lascarides. 2003. *Logics of conversation*. Cambridge University Press.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Zineb Bennis, Julie Hunter, and Nicholas Asher. 2023. A simple but effective model for attachment in discourse parsing with multi-task learning for relation labeling. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3404–3409.

Claire Bonial, Mitchell Abrams, David Traum, and Clare Voss. 2021. Builder, we have done it: evaluating & extending dialogue-AMR NLU pipeline for two collaborative domains. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 173–183.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020. Dialogue-AMR: abstract meaning representation for dialogue. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 684–695.

Julia Bonn, Martha Palmer, Jon Cai, and Kristin Wright-Bettner. 2020. Spatial AMR: Expanded spatial annotation in the context of a grounded minecraft corpus. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020),*.

Barbara G Deutsch, Grosz. 1974. *The structure of task oriented dialogs*. Artificial Intelligence Center, SRI International.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

K. Forbes, E. Miltsakaki, R. Prasad, A. Sarkar, A. K. Joshi, and B. L. Webber. 2003. D-LTAG system: Discourse parsing with a lexicalized tree-adjoining grammar. *Journal of Logic, Language and Information*, 12(3):261–279.

Barbara J Grosz and Candace L Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational linguistics*, 12(3):175–204.

Julie Hunter, Nicholas Asher, and Alex Lascarides. 2018. A formal semantics for situated conversation. *Semantics and Pragmatics*, 11. DOI: http://dx.doi.org/10.3765/sp.11.10.

Prashant Jayannavar, Anjali Narayan-Chen, and Julia Hockenmaier. 2020. Learning to execute instructions in a Minecraft dialogue. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2589–2602, Online. Association for Computational Linguistics.

Jiaqi Li, Ming Liu, Min-Yen Kan, Zihao Zheng, Zekun Wang, Wenqiang Lei, Ting Liu, and Bing Qin. 2020. Molweni: A challenge multiparty dialogues-based machine reading comprehension dataset with discourse structure. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2642–2652.

W. Mann and S. Thompson. 1987. Rhetorical structure theory : a theory of text organization. Technical report, Information Science Institute.

Y. Mathet and A. Widlöcher. 2009. La plate–forme GLOZZ : environnement d’annotation et d’exploration de corpus. In *Traitement automatique des langues naturelles*.

Mathieu Morey, Philippe Muller, and Nicholas Asher. 2018. A dependency perspective on RST discourse parsing and evaluation. *Computational Linguistics*, 44(2):197–235.

Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. Collaborative dialogue in Minecraft. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5405–5415, Florence, Italy. Association for Computational Linguistics.

Jérémy Perret, Stergos Afantenos, Nicholas Asher, and Mathieu Morey. 2016. Integer linear programming for discourse parsing. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 99–109, San Diego, California. ACL.

L. Polanyi, C. Culy, M. van den Berg, G. L. Thione, and D. Ahn. 2004. A rule based approach to discourse parsing. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 108–117, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber. 2008. The Penn Discourse Treebank 2.0. In *Proceedings of LREC'08*, Marrakech, Morocco. ELRA. Http://www.lrec-conf.org/proceedings/lrec2008/.

Zhengxiang Shi, Yue Feng, and Aldo Lipani. 2022. Learning to execute actions or ask clarification questions. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2060–2070.

Zhouxing Shi and Minlie Huang. 2019. A deep sequential model for discourse parsing on multiparty dialogues. In *AAAI*.

Rajen Subba and Barbara Di Eugenio. 2009. An effective discourse parser that uses rich linguistic information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 566–574.

Ante Wang, Linfeng Song, Hui Jiang, Shaopeng Lai, Junfeng Yao, Min Zhang, and Jinsong Su. 2021. A structure self-aware model for discourse parsing on multi-party dialogues. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3943–3949. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Bonnie Webber, Norman Badler, Barbara Di Eugenio, Chris Geib, Libby Levison, and Michael Moore. 1995. Instructions, intentions and expectations. *Artificial Intelligence*, 73(1-2):253–269.

F. Wolf and E. Gibson. 2005. Representing discourse coherence: A corpus based study. *Computational Linguistics*, 31(2):249–287.

Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, Brian Ichter, Ted Xiao, Peng Xu, Andy Zeng, Tingnan Zhang, Nicolas Heess, Dorsa Sadigh, Jie Tan, Yuval Tassa, and Fei Xia. 2023. Language to rewards for robotic skill synthesis. *Arxiv preprint arXiv:2306.08647*.