

MUDiPA: A Human-in-the-Loop AI-Assisted Platform for Discourse Graph Annotation in Multi-party Dialogues

Laura De Santis^{1*}, Gaetano Cimino^{1*}, Nils Feldhus²,
Giuseppe Carenini³, Vincenzo Deufemia¹

¹Department of Computer Science, University of Salerno, Italy

²Center for Language and Cognition, University of Groningen, The Netherlands

³Department of Computer Science, University of British Columbia, Canada

{ldesantis, gcimino, deufemia}@unisa.it
n.feldhus@rug.nl carenini@cs.ubc.ca

Abstract

Annotating discourse structure in multi-party dialogues is a cognitively demanding task that requires specialized expertise, yet existing annotation tools are either fully manual or provide only limited assistance. We present MUDiPA, a web-based, human-in-the-loop platform for discourse graph annotation in multi-party dialogues, built around two plug-gable AI backends: a suggestion engine that produces confidence-ranked arc proposals, and an LLM-based reasoning engine that generates on-demand natural-language rationales for candidate attachments and relation labels. In a counterbalanced, within-subjects study with eight annotators, we compare AI-assisted annotation with MUDiPA against manual annotation. The AI-assisted condition achieves higher inter-annotator agreement on relation labels, higher accuracy against the gold standard, and lower perceived workload, while requiring only a modest increase in annotation time. These findings indicate that the principal value of LLMs in discourse annotation lies in supporting human decision-making, improving both annotation quality and inter-annotator agreement relative to unassisted annotation.¹

1 Introduction

Understanding discourse structure in multi-party dialogues is essential for tasks such as summarization (Feng et al., 2022), dialogue comprehension (Ma et al., 2023), and action prediction (Chaturvedi et al., 2024). Discourse parsing aims to transform a conversation into a structured representation by identifying semantic and pragmatic relations between *Elementary Discourse Units* (EDUs).

These discourse structures are commonly represented using the *Segmented Discourse Representation Theory* (SDRT) framework (Asher and Las-

carides, 2005), which models dialogues as dependency graphs whose nodes correspond to EDUs and whose labeled edges encode discourse relations. Figure 1 illustrates examples of SDRT relations between EDUs, including *Result*, *Question-Answer Pair* (QAP), *Contrast*, and *Continuation*.

Despite established annotation frameworks, progress in multi-party dialogue discourse parsing remains limited by the scarcity of annotated corpora. Existing SDRT-style resources, including the Strategic Conversations corpus (STAC) (Asher et al., 2016) and Molweni (Li et al., 2020), are small, domain-specific, and largely monolingual; creating them requires expert knowledge and substantial manual effort, making it costly to extend these resources to new domains, languages, or modalities (Shukla et al., 2025; Liu et al., 2026).

The annotation task is cognitively demanding because annotators must identify long-range dependencies across multiple speakers while disambiguating fine-grained discourse relations (López Cortez and Jacobs, 2023; Cimino et al., 2024). These challenges increase effort and reduce consistency, particularly when multiple relation labels are plausible for the same pair of EDUs (Scholman et al., 2022; Pyatkin et al., 2023).

Most existing tools used for constructing discourse-annotated corpora require annotators to perform the entire process manually (Widlöcher and Mathet, 2012; Li, 2023). In these settings, users are responsible for inspecting the dialogue and explicitly identifying and labeling discourse relations between EDUs without structured computational support. While this approach ensures full control over the annotation process, it does not alleviate the cognitive burden associated with long-distance reasoning and relation disambiguation described above.

Recent work has explored the use of Large Language Models (LLMs) for discourse annotation (Petukhova and Kochmar, 2025). However, LLMs

^{*}These authors contributed equally to this work.

¹Screen recording, source code, and installation instructions are available at [MuDiPA](#).

remain unreliable for automatic annotation in multi-party dialogue discourse parsing, particularly for rare discourse relations and long-distance dependencies (Li and Carenini, 2026). Other approaches improve downstream parsers through LLM-based dialogue augmentation (Cimino et al., 2026) but still depend on manually annotated corpora.

To address these limitations, we present MUDiPA (**M**ulti-party **D**ialogue **P**arsing **A**ssistant), a web-based platform for dialogue discourse annotation that adopts a human-AI collaborative workflow (Amershi et al., 2019). The system combines predictions from state-of-the-art discourse parsers with LLM-generated explanations to assist annotators in inspecting and refining discourse structures.

The contributions of this work are threefold. First, we present a web-based platform for dialogue discourse annotation. Second, we introduce a modular assistance framework comprising two interchangeable components: a suggestion engine that provides confidence-ranked predictions of discourse links and relation labels, and a reasoning engine that generates natural-language rationales for candidate annotations. Third, we evaluate the platform through a user study with eight annotators, showing that AI-assisted annotation improves inter-annotator agreement on relation labels and alignment with gold-standard annotations, reduces perceived workload, and incurs only a modest, non-significant increase in annotation time.

2 Related Systems

Discourse-oriented tools have long been used to construct dialogue annotation datasets. Glozz (Widlöcher and Mathet, 2012), one of the most widely adopted platforms for discourse annotation, was used to construct corpora such as STAC (Asher et al., 2016) and MSDC (Thompson et al., 2024b). More recently, SDRT-Anno (Li, 2023) was proposed specifically for SDRT-style dialogue discourse annotation, providing an interface tailored to this annotation framework. Despite their domain-specific design, both platforms rely entirely on manual annotation and provide neither automated suggestions nor AI-assisted support for attachment decisions or relation assignment.

General-purpose annotation platforms have also been adopted for dialogue discourse annotation. For example, INCEpTION (Klie et al., 2018) was recently used to annotate the CoMuMDR corpus

(Shukla et al., 2025). While INCEpTION supports model-generated annotation suggestions, its assistance is limited to prediction recommendations and does not provide reasoning support or discourse-specific guidance during graph construction.

At the other end of the spectrum, recent work has explored fully automatic discourse annotation. For example, the pipeline proposed by Petukhova and Kochmar (2025) employs LLMs to generate discourse annotations without human involvement. However, because off-the-shelf LLMs used as dialogue discourse parsers still exhibit substantial limitations (Li and Carenini, 2026), fully automatic annotations are not yet suitable for constructing gold-standard corpora.

MUDiPA bridges the gap between these approaches by combining parser-based suggestions with LLM-generated natural-language rationales within an interactive human-in-the-loop workflow. Rather than replacing the annotator, the system provides confidence-ranked suggestions together with explicit explanations that help users inspect, compare, and critically evaluate alternative discourse interpretations while ultimately retaining full control over the final annotation decisions (Horvitz, 1999; Amershi et al., 2019).

3 System Overview

MUDiPA is a web-based platform for discourse structure annotation.² The annotator works on an unlabeled, pre-segmented dialogue, where EDUs are provided in advance and must be connected by discourse relations. To address challenges related to long-distance dependencies and ambiguous discourse relations, MUDiPA supports a unified *draft* $\rightarrow$ *reason* $\rightarrow$ *validate* workflow, assisted by two complementary AI components integrated into the annotation process. The *suggestion engine* reduces search effort by proposing ranked candidates for EDU attachments and relation labels, whereas the *reasoning engine* supports decision making in ambiguous cases by providing explanatory feedback on candidate links or relations.

Interface and Annotation Workflow. As illustrated in Figure 1, the annotation workflow is implemented in a single web interface. The annotator selects the corpus and a dialogue from the top bar (A). The selected dialogue is displayed through a *dialogue list* (B) showing EDUs grouped

²Implementation details are provided in Appendix A.

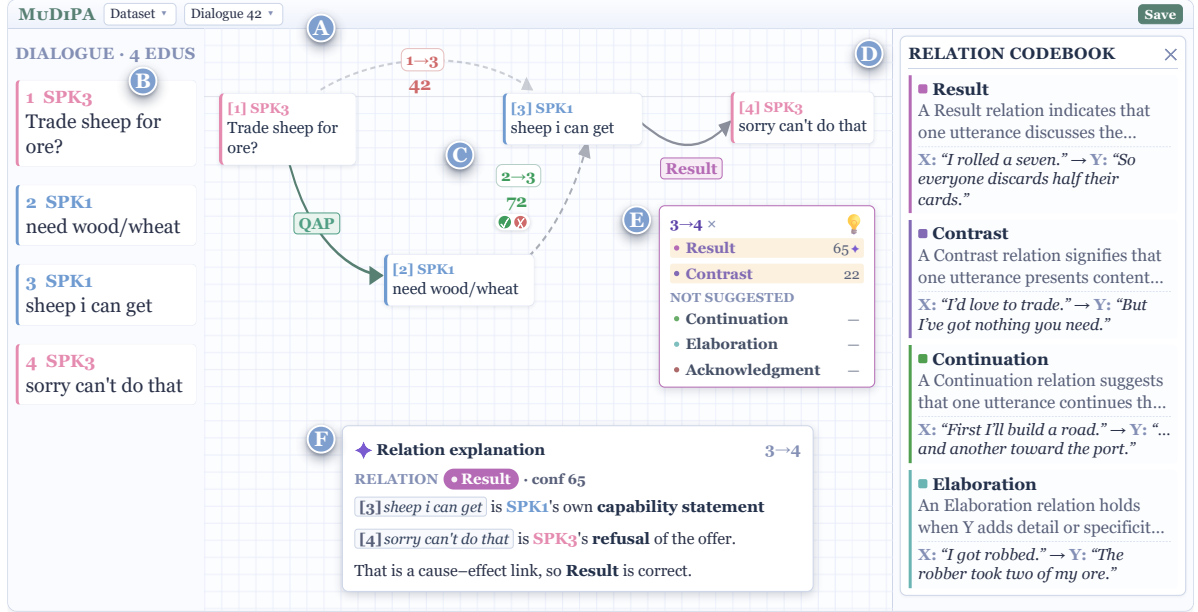


Figure 1: The MUDiPA interface illustrated through the annotation of a dialogue from the STAC corpus. **A Load.** The annotator selects the STAC corpus and the target dialogue to be annotated. **B Dialogue list.** The whole dialogue at a glance, with one card per EDU, color-coded by speaker. **C Link suggestion.** After selecting EDU 3, the parser proposes candidate incoming links. The system suggests the attachment from EDU 2 with the highest confidence (72), while the direct attachment from EDU 1 receives a lower confidence (42). The annotator can accept or reject the suggestions. **D Relation codebook.** Once an attachment is confirmed, the annotator can inspect the SDRT relation inventory, where each relation is accompanied by a concise definition and illustrative example. **E Relation suggestion.** For the created link from EDU 3 to EDU 4, the parser ranks candidate discourse relations, predicting *Result* (confidence 65) ahead of *Contrast* (22), while the remaining relations are available through the codebook. **F Reasoning.** Upon request, the reasoning engine justifies the proposed *Result* relation by explaining that SPK1’s statement (“sheep i can get”) leads to SPK3’s refusal (“sorry can’t do that”), providing evidence that the selected discourse relation is appropriate. A similar rationale can also be requested for the attachment alone, i.e., to justify whether two EDUs should be linked, independently of the relation label.

by speaker, while the annotation canvas provides a pan-and-zoom view of the discourse graph. Working through the dialogue, the annotator selects an EDU and requests link suggestions (💡): the candidate attachment points appear as dashed *arc proposals* with their confidence scores (🔍), which can be accepted or rejected in one click. Exposing model confidence to guide the validation of suggestions follows established practice in annotation support (Klie et al., 2018). Once an arc is placed, the annotator assigns its relation label from the *relation codebook* (📖), which provides immediate access to the definition and a canonical example of each relation; optionally, a side panel provides the suggestion engine’s confidence ranking of candidate labels (📊). In cases of uncertainty, the annotator can query the LLM with the reasoning functionality (💡), which evaluates the candidate link/relation given the preceding context and returns a rationale for the suggested annotation (📝). The annotator then validates the arc and label, incorporating the

confirmed relation into the final annotation.

Sub-dialogue Navigation. To facilitate navigation in long or highly threaded dialogues, we adopt the notion of sub-dialogues introduced in Cimino et al. (2024), which captures coherent conversational segments. Building on this definition, the *sub-dialogue explanation* module (Figure 2) groups EDUs into conversational sub-threads based on reply structure and topical coherence. In addition, annotators can request LLM-generated explanations of individual conversational flows.

Multimodal Support. Although MUDiPA is primarily intended for text-based discourse annotation, its interface also enables the inspection of multimodal dialogues. To accommodate emerging multimodal discourse resources (Gong et al., 2024; Shukla et al., 2025; Liu et al., 2026), the system integrates synchronized visualization of video, audio, and textual content, as illustrated in Figure 3.

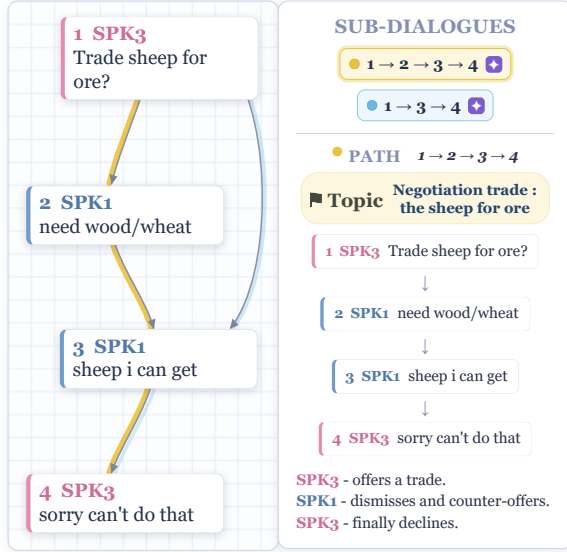


Figure 2: Sub-dialogue navigation. MUDiPA groups EDUs into coherent sub-dialogues shown as color-coded paths in the discourse graph (left). The side panel (right) lists the identified sub-dialogues and displays an on-demand explanation for the selected one.

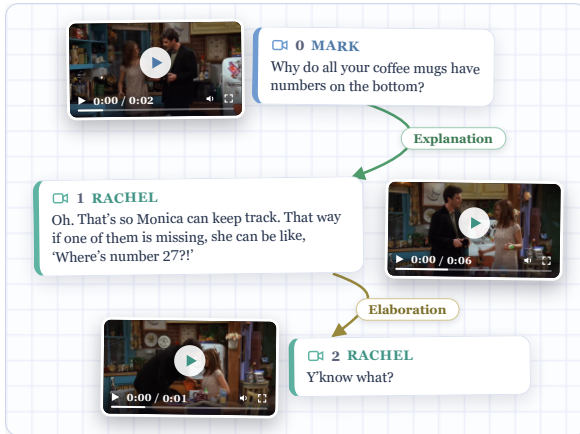


Figure 3: Multimodal view supporting synchronized video, speech, and text during discourse annotation.

4 Evaluation

4.1 Experimental Setup

Study Design. Following prior work on LLM-assisted annotation platforms (Ningtyas et al., 2026), we evaluated the effectiveness of MUDiPA by comparing discourse annotation performed with LLM assistance against manual annotation. The study involved 2 researchers, 2 doctoral students, and 4 master’s students with expertise in discourse analysis. The experiment adopted a counterbalanced within-subjects design and was conducted across two sessions held on separate days. Each participant completed two annotation sessions, one under the manual condition and one with LLM as-

sistance, annotating 15 dialogues per session for a total of 30 dialogues. Participants were randomly assigned, in groups of four, to one of two annotation sequences. Group *AB* annotated *Set 1* manually and *Set 2* with assistance, whereas Group *BA* followed the reverse order. This design controls for participant, order, and dialogue effects, allowing observed differences to be attributed to the annotation condition. In the manual condition, the two AI backends were disabled. Annotators used the segmented dialogue and arc-drawing interface, together with the SDRT relation codebook and the per-arc audit log, but received no suggestions or rationales. In the assisted condition, the suggestion and reasoning engines were enabled within the same interface.

The 30 dialogues were randomly sampled from the STAC corpus and stratified by dialogue length into 10 short (< 10 EDUs), 10 medium (10–19 EDUs), and 10 long (≥ 20 EDUs) dialogues to investigate whether the impact of LLM assistance varies with dialogue complexity.

Before starting the annotation tasks, all participants completed a training session covering the fundamentals of SDRT, a detailed presentation of the 16 discourse relation types defined in STAC, and an overview of the Settlers of Catan game mechanics underlying the corpus. After each annotation session, participants completed the NASA Task Load Index (NASA-TLX) questionnaire to assess perceived workload. Upon completing the study, they also completed the System Usability Scale (SUS) questionnaire to evaluate the overall usability of MUDiPA.

Evaluation Metrics. We report the micro-averaged *Link* F_1 score, together with its corresponding precision and recall values, where an arc is considered correct if its head matches the gold attachment regardless of the relation label. We also report the micro-averaged *Labeled* F_1 score, along with its precision and recall, which requires both the head and the relation label to match the STAC gold annotation. Furthermore, we measure inter-annotator agreement (IAA) using the same evaluation metrics, computed between annotators rather than with respect to the gold annotation. Specifically, for each dialogue and experimental condition, we calculate pairwise Link and Labeled F_1 scores for all annotator pairs and average the resulting values. Finally, we evaluate the annotation process by measuring the average time required to

annotate a dialogue, the number of reasoning and suggestion requests for links and relation labels, the acceptance rate of the suggestion engine’s proposals, the participants’ perceived workload through the NASA-TLX questionnaire, and the usability of MUDiPA through the SUS questionnaire. Statistical significance is assessed using paired Wilcoxon signed-rank tests, conducted per participant for accuracy and workload measures and per dialogue for IAA. We apply the Holm–Bonferroni correction to account for multiple comparisons.

Model Selection. We intentionally evaluate MUDiPA in a cold-start setting, where annotation must be performed on a previously unseen domain without access to labeled target-domain data. Accordingly, we employ an out-of-domain parser for the suggestion engine, following Liu et al. (2026). In particular, we test different state-of-the-art fine-tuned parsers alongside out-of-the-box LLMs on another well-known corpus, namely Molweni (Li et al., 2020). Among the methods considered, DDPE (Liu et al., 2025) achieves the best performance, with a Link F_1 of 87.6 and a Labeled F_1 of 62.9. Accordingly, we select this parser as the backbone of the suggestion engine and derive confidence scores from its predicted probability distributions. For the reasoning engine, we adapt the explanation prompt introduced by Liu et al. (2025) and employ the Claude Opus 4.8 model to generate annotation rationales.³ However, this choice is not restrictive, as the proposed tool can be instantiated with any other LLM.

4.2 Results

Annotation quality. Table 1 presents precision, recall, and micro-averaged F_1 scores computed against the STAC gold annotations and aggregated across all 30 dialogues. The assisted condition yields statistically significant improvements over the unassisted condition for both link prediction and relation labeling. Specifically, Link F_1 increases from 72.0 to 77.9 ($p = 0.039$), while Labeled F_1 improves from 56.1 to 64.9 ($p = 0.032$). Improvements are observed in both precision and recall. For Labeled prediction, precision increases from 59.0 to 68.4, while recall rises from 53.4 to 61.7, indicating that assistance improves both the completeness and correctness of discourse annota-

³The prompts used by the reasoning engine are provided in Appendix B. A human evaluation of the quality of the generated rationales is described in Appendix C.

		Link			Labeled			Time (s)
		P	R	F_1	P	R	F_1	
AUTOMATIC	Raw parser	53.2	63.4	57.9	16.5	19.6	17.9	—
	LLM-only	75.9	69.3	72.5	43.0	39.3	41.1	—
	Parser+LLM	74.4	68.2	71.1	43.5	39.9	41.6	—
HUMAN SUPERVISED	Manual	75.8	68.6	72.0	59.0	53.4	56.1	200
	MUDiPA	82.1	74.0	77.9	68.4	61.7	64.9	260
	gain (Δ)	+6.3	+5.4	+5.9*	+9.4	+8.3	+8.8*	+60

Table 1: Performance comparison in terms of precision (P), recall (R), and micro-averaged F_1 score against the STAC gold annotations, aggregated over the 30 dialogues. The table also reports the mean annotation time per dialogue (in seconds). Times for the LLM-assisted condition include engine latency. Statistical significance is assessed using a Wilcoxon signed-rank test against Manual (* $p < 0.05$); the difference in annotation time is not statistically significant. Reported p -values are Holm–Bonferroni corrected.

tions. For comparison, the table also reports the performance of the suggestion engine evaluated independently on the same dialogues. The out-of-domain parser alone achieves moderate performance on link prediction but substantially lower performance on relation labeling (Link/Labeled F_1 57.9/17.9); the assisted condition exceeds it on both metrics, demonstrating the benefit of combining automatic suggestions with human validation. We further report two fully-automatic baselines that use Claude Opus 4.8, the same model as the reasoning engine: the LLM parsing each dialogue on its own (*LLM-only*), and the LLM revising the parser’s draft (*Parser+LLM*). Both match unaided human performance on attachment (Link F_1 72.5 and 71.1) but fall well short on relation labeling (41.1 and 41.6 vs. 64.9 for the assisted condition), and pairing the parser with the LLM does not improve over the LLM alone. This indicates that the benefit of assistance lies in human validation of fine-grained relations rather than in full automation.

Agreement. Table 2 reports IAA results. Agreement on relation labels improves substantially when using MUDiPA (pairwise Labeled F_1 : 49.2 $\rightarrow$ 60.3, $p = 0.004$), whereas agreement on attachment remains essentially unchanged (79.2 $\rightarrow$ 78.5, $p = 0.95$). These findings suggest that the assistance primarily benefits the decisions where annotators are most likely to disagree, namely the selection among fine-grained relation labels that

	Link	Labeled
Manual	79.2 $\pm$ 11.5	49.2 $\pm$ 13.1
MUDiPA	78.5 $\pm$ 13.8	60.3 $\pm$ 14.4
IAA gain (Δ)	-0.7	+11.1*

Table 2: Inter-annotator agreement (pairwise F_1 , mean $\pm$ sd). Statistical significance is assessed using the Wilcoxon signed-rank test (* $p < 0.05$). Reported p -values are Holm–Bonferroni corrected.

are inherently difficult to distinguish.

Where and how assistance helps. To better understand the observed improvements, we analyze how annotators interacted with the assistance mechanisms. Results are reported in Table 3. We first examine *what* annotators requested assistance for. Suggestion requests are distributed across both annotation decisions (44% for links and 56% for relation labels), whereas reasoning requests are concentrated almost exclusively on relation labeling (97%). Although request frequencies alone do not determine task difficulty, together with the lower acceptance rate for label suggestions, the larger gains in labeled accuracy and agreement, and the concentration of reasoning requests, these interaction logs are consistent with relation labeling being the primary source of annotation uncertainty. *Link prediction* is largely handled by the suggestion engine: annotators keep 80% of its top-two attachment suggestions, and agreement is high and essentially unchanged (79.2 $\rightarrow$ 78.5, Table 2). *Relation labeling* is substantially more challenging: annotators keep only 25% of the parser suggestions and turn to the reasoning engine on a subset of queried arcs reflecting annotator-selected cases of uncertainty. On these arcs, the parser’s top-1 relation is incorrect in 69% of cases, while the reasoning engine identifies these parser errors with precision 86.1%, recall 92.5%, and balanced accuracy 79.6%. Its behaviour is that of an error *detector* rather than a *labeller*: flagging that a proposed relation is wrong is easier than generating the correct one from scratch, which is why the LLM on its own reaches only 41.1 Labeled F_1 (Table 1). The engine therefore narrows the annotator’s decision rather than replacing it.

The effects of assistance scale with dialogue complexity, as illustrated in Table 4. Assistance requests grow with length, and the attachment gain is largest on long dialogues (Δ Link F_1 +7.1), where the candidate attachment space is larger; the labeling gains in both accuracy and agreement re-

Assistance usage	Link	Label
Suggestion requests	44% (478/1094)	56% (616/1094)
Accepted	80% (381/478)	25% (154/616)
Reasoning requests	3% (12/468)	97% (456/468)

Reasoning engine	P	R	Bal. acc.
Parser error detection	86.1	92.5	79.6

Table 3: Assistance usage by decision type. Counts (percentages) are reported for suggestion and reasoning requests. Percentages are computed over the total requests of the corresponding assistance type, whereas acceptance rates are computed over the corresponding suggestion requests. The parser’s top-1 relation is incorrect on 69% of the queried arcs; the bottom table reports the reasoning engine’s detection of these parser errors. As this error prevalence is high, we report balanced accuracy alongside precision and recall; at 79.6% the engine remains well above chance, so its detection is not an artefact of the class imbalance.

	F ₁ gain		IAA gain		Requests/dial.		
Difficulty	Δ Link	Δ Lab	Δ Link	Δ Lab	Link	Rel.	Reas.
Short	+4.5	+10.1	-0.2	+12.5	2.2	2.9	1.6
Medium	+3.5	+13.8	-4.2	+9.8	3.3	4.4	3.4
Long	+7.1	+6.3	+4.6	+7.3	6.5	8.1	6.7

Table 4: Effect of assistance by dialogue length: micro-averaged F_1 and IAA *gains* of MUDiPA over Manual, and mean engine requests per dialogue (link/relation suggestions and relation reasoning).

main substantial at every length (Δ Labeled F_1 up to +13.8, Δ IAA up to +12.5).

Cost and workload. The mean annotation time per dialogue, reported in Table 1, was 200 s in the Manual condition and increased to 260 s when using MUDiPA. One possible explanation is that annotators spent additional time reviewing parser suggestions and consulting LLM rationales, which is consistent with the increase in assistance requests as dialogue length grows. This observation aligns with prior studies showing that LLM-assisted annotation does not necessarily reduce annotation time (Schroeder et al., 2025; Ningtyas et al., 2026).

Despite the longer annotation process, participants reported a lower perceived workload, as shown in Table 5. The weighted NASA-TLX score decreased significantly from 66.5 to 57.4 ($p = 0.008$). This reduction was consistent across all workload dimensions included in the analysis. Physical demand was not analyzed individually, as it is not a meaningful factor in computer-based annotation tasks. The largest reductions were ob-

Condition	NASA-TLX						SUS
	M	T	P	E	F	Overall	Score
Manual	75.8	74.9	58.4	73.8	70.1	66.5	70.6
MUDiPA	63.8	57.4	49.0	60.5	58.9	57.4	
Δ	12.0*	17.5*	9.4*	13.3*	11.2*	9.1*	

Table 5: NASA-TLX workload ratings across conditions and SUS usability scores for MUDiPA. NASA-TLX dimensions: M = Mental, T = Temporal, P = Performance, E = Effort, F = Frustration. Significance is assessed using the Wilcoxon signed-rank test (* $p < 0.05$). Reported p -values are Holm–Bonferroni corrected.

served in temporal demand (74.9 $\rightarrow$ 57.4) and mental demand (75.8 $\rightarrow$ 63.8), followed by frustration (70.1 $\rightarrow$ 58.9), effort (73.8 $\rightarrow$ 60.5), and perceived performance (58.4 $\rightarrow$ 49.0), with all differences reaching statistical significance ($p < 0.05$). The SUS yielded a mean score of 70.6, indicating acceptable usability and approaching the threshold commonly associated with “good” usability (Bangor et al., 2009). However, the relatively large standard deviation suggests considerable variability in participants’ perceptions of the system’s usability.

5 Conclusion

We presented MUDiPA, a human-in-the-loop platform for discourse structure annotation in multi-party dialogues that combines parser-based suggestions with LLM-generated reasoning in a unified workflow. Our user study demonstrates that AI-assisted annotation improves both annotation quality and IAA on relation labels while reducing perceived workload, with only a modest increase in annotation time. Comparisons with fully automatic baselines further show that, in the evaluated setting, automation alone falls short of the quality achieved through human–AI collaboration, particularly for fine-grained relation labeling. These findings suggest that the value of AI assistance in discourse annotation lies primarily in supporting human decision-making rather than replacing human annotators. Because the present study evaluates the combined workflow, future work will investigate the individual contributions of suggestion and reasoning components, adaptive assistance strategies, and support for additional discourse formalisms and annotation settings.

Limitations

The execution of LLM API calls within the reasoning engine incurs both monetary costs and an associated energy footprint. However, since expla-

nations are generated on demand, upon user interaction, rather than being pre-computed for every candidate arc, MUDiPA’s architecture limits these costs to instances of actual annotator uncertainty, rather than incurring them across the entire space of possible attachments.

The reasoning engine generates natural language rationales designed to assist annotators rather than replace their judgment. Accordingly, these rationales should not be interpreted as authoritative explanations or definitive justifications. Since rationales are generated from input dialogue content, they may reproduce sensitive material or reflect biases from the underlying LLM. MUDiPA does not currently perform automatic content moderation; therefore, deployers should apply appropriate safeguards and review generated rationales before their inclusion in shared or published materials. Furthermore, the suggestion engine was evaluated out-of-domain, and its performance may vary with different parsers or in-domain training data.

Because all annotators consult the same reasoning engine, part of the agreement gain may reflect convergence on its shared advice rather than independent agreement, so corpora produced in this way may inherit the model’s biases. The possibility of training-data contamination further complicates the interpretation of the observed agreement. Given that STAC has long been publicly available, it may have been incorporated into the reasoning model’s pretraining corpus. Although the use of an out-of-domain parser and the annotators’ selective reliance on model suggestions mitigate this concern, they cannot rule out the possibility of contamination.

Ethical Considerations

MUDiPA is intended for annotating research corpora released under appropriate licenses. For proprietary or newly collected dialogue data, administrators must ensure data rights and, when required, participant consent before upload. Since MUDiPA does not provide automatic Personally Identifiable Information (PII) detection or redaction, deployers should apply appropriate safeguards before uploading data to the system. Multimodal data may introduce additional privacy risks, such as voice or facial identifiability, which should be addressed through suitable data handling practices.

References

- Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. [Guidelines for Human-AI Interaction](#). In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–13. Association for Computing Machinery.
- Nicholas Asher, Julie Hunter, Mathieu Morey, Farah Benamara, and Stergos Afantenos. 2016. [Discourse structure and dialogue acts in multiparty dialogue: the STAC corpus](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2721–2727. European Language Resources Association (ELRA).
- Nicholas Asher and Alex Lascarides. 2005. *Logics of Conversation*. Studies in natural language processing. Cambridge University Press.
- Aaron Bangor, Philip Kortum, and James Miller. 2009. [Determining what individual sus scores mean: adding an adjective rating scale](#). *Journal of Usability Studies*, 4(3):114–123.
- Akshay Chaturvedi, Kate Thompson, and Nicholas Asher. 2024. [Nebula: A discourse aware Minecraft builder](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6431–6443. Association for Computational Linguistics.
- Gaetano Cimino, Giuseppe Carenini, and Vincenzo Deufemia. 2026. [MIMIC: Multi-party dialogue augmentation via speaker stylistic transfer](#). In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 2693–2719. Association for Computational Linguistics.
- Gaetano Cimino, Chuyuan Li, Giuseppe Carenini, and Vincenzo Deufemia. 2024. [Coherence-based dialogue discourse structure extraction using open-source large language models](#). In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 297–316. Association for Computational Linguistics.
- Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2022. [A survey on dialogue summarization: Recent advances and new frontiers](#). In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5453–5460. International Joint Conferences on Artificial Intelligence Organization. Survey Track.
- Chen Gong, Dexin Kong, Suxian Zhao, Xingyu Li, and Guohong Fu. 2024. [MODDP: A multi-modal open-domain chinese dataset for dialogue discourse parsing](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10561–10573.
- Eric Horvitz. 1999. [Principles of mixed-initiative user interfaces](#). In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, page 159–166. Association for Computing Machinery.
- Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. [The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation](#). In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics.
- Chuyuan Li. 2023. [SDRT-Anno: a web platform for SDRT-style dialogue annotation](#). Web tool, no associated paper.
- Chuyuan Li and Giuseppe Carenini. 2026. [BeDiscover: The benchmark of discourse understanding in the era of reasoning language models](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4417–4479. 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. International Committee on Computational Linguistics.
- Shannan Liu, Peifeng Li, Yaxin Fan, and Qiaoming Zhu. 2025. [Enhancing multi-party dialogue discourse parsing with explanation generation](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1531–1544. Association for Computational Linguistics.
- Shannan Liu, Peifeng Li, Yaxin Fan, and Qiaoming Zhu. 2026. [DraDDP: A multimodal multi-party dialogue discourse parsing dataset](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 27359–27373. Association for Computational Linguistics.
- S. Magalí López Cortez and Cassandra L. Jacobs. 2023. [Incorporating annotator uncertainty into representations of discourse relations](#). In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 530–537. Association for Computational Linguistics.
- Xinbei Ma, Zhuosheng Zhang, and Hai Zhao. 2023. [Enhanced speaker-aware multi-party multi-turn dialogue comprehension](#). *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2410–2423.
- Annisa Maulida Ningtyas, Guntur Budi Herwanto, Yunita Sari, Rifki Afina Putri, Filip Kovacevic, Alaa El-Ebshihy, Varvara Arznt, and Florina Piroi. 2026. [AnnoHID: LLM-assisted annotation framework for low-resource medical texts](#). In *Proceedings of the*

64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 683–691. Association for Computational Linguistics.

Kseniia Petukhova and Ekaterina Kochmar. 2025. [A fully automated pipeline for conversational discourse annotation: Tree scheme generation and labeling with large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 15829–15852. Association for Computational Linguistics.

Valentina Pyatkin, Frances Yung, Merel C. J. Scholman, Reut Tsarfaty, Ido Dagan, and Vera Demberg. 2023. [Design choices for crowdsourcing implicit discourse relations: Revealing the biases introduced by task design](#). *Transactions of the Association for Computational Linguistics*, 11:1014–1032.

Merel Scholman, Tianai Dong, Frances Yung, and Vera Demberg. 2022. [DiscoGeM: A crowdsourced corpus of genre-mixed implicit discourse relations](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3281–3290. European Language Resources Association.

Hope Schroeder, Deb Roy, and Jad Kabbara. 2025. [Just put a human in the loop? investigating LLM-assisted annotation for subjective tasks](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25771–25795. Association for Computational Linguistics.

Divyaksh Shukla, Ritesh Baviskar, Dwijesh Gohil, Aniket Tiwari, Atul Shree, and Ashutosh Modi. 2025. [CoMuMDR: Code-mixed multi-modal multi-domain corpus for discourse paRsing in conversations](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10834–10849. Association for Computational Linguistics.

Kate Thompson, Akshay Chaturvedi, Julie Hunter, and Nicholas Asher. 2024a. [Llamipa: An incremental discourse parser](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6418–6430. Association for Computational Linguistics.

Kate Thompson, Julie Hunter, and Nicholas Asher. 2024b. [Discourse structure for the Minecraft corpus](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4957–4967. ELRA and ICCL.

Antoine Widlöcher and Yann Mathet. 2012. [The glozz platform: a corpus annotation and mining tool](#). In *Proceedings of the 2012 ACM Symposium on Document Engineering, DocEng '12*, page 171–180. Association for Computing Machinery.

A Implementation Details

MUDiPA is a self-hostable web application comprising a Python/Flask backend and a React-based frontend that renders the pan-and-zoom canvas as an SVG. The two AI engines are exposed through lightweight *scoring* and *reasoning* interfaces accessible via REST and SDK layers, enabling them to be independently replaced, mocked, or deployed on separate infrastructure without affecting the annotation logic. Dialogues, EDUs, arcs, and interaction events are stored in a relational database, while each arc modification is recorded as an individual log entry; this design supports undo functionality and automatic saving. Completed annotations can be exported in JSON and Glozz formats to ensure compatibility with existing evaluation pipelines.

B Reasoning Engine Prompt

This appendix presents the three prompt templates used to query the reasoning engine (Claude Opus 4.8), each adapted from the Explanation Generation (EGM) prompt proposed by Liu et al. (2025). These include the **Link Explanation Prompt**, which evaluates whether a candidate arc should be established independently of any relation label; the **Relation Explanation Prompt**, which assesses the appropriateness of the label assigned to a confirmed arc; and the **Sub-dialogue Explanation Prompt**, which generates both a topic label and a narrative explanation for a given sub-dialogue path. To facilitate more contextually grounded explanations, the first two prompts incorporate the 15 preceding EDUs rather than only the target EDU pair, following the approach adopted by Thompson et al. (2024a) for the prediction task.

Link Explanation Prompt

Given the following dialogue context and candidate arc, assess whether the target EDU should be linked to the source EDU, and only output the verdict and a one-sentence explanation referring to the numbered EDUs. Don't add anything else.

Dialogue context (last 15 utterances, numbered):

```
{context}
```

Candidate arc:

```
[{source_idx}]->[{target_idx}]
[{source_idx}] {source_speaker}:
{source_edu}
[{target_idx}] {target_speaker}:
{target_edu}
```

Relation Explanation Prompt

Given the following relation inventory, dialogue context and confirmed arc with its proposed relation, assess whether the relation label is appropriate, and only output the verdict and a one-sentence explanation referring to the numbered EDUs. Don't add anything else.

Relation inventory:

{relation_inventory}

Dialogue context (last 15 utterances, numbered):

{context}

Confirmed arc:

[{source_idx}]-->[{target_idx}]

[{source_idx}] {source_speaker}:

{source_edu}

[{target_idx}] {target_speaker}:

{target_edu}

Proposed relation:

'{proposed_relation}'

Sub-dialogue Explanation Prompt

Given the following dialogue context, provide a short topic label and a one-sentence explanation of the conversational flow it captures, referring to the numbered EDUs in order, and only output this content. Don't add anything else.

Dialogue context:

{path}

EDUs in path (numbered):

{path_edus}

C Rationale Quality

We evaluate the rationales generated by the reasoning engine along two dimensions: *Faithfulness* and *Helpfulness*. *Faithfulness* assesses whether every claim presented in a rationale is directly supported by the dialogue, thereby ensuring that the explanation is free from hallucinated or fabricated content. *Helpfulness* measures the extent to which a rationale provides relevant and actionable evidence that justifies the proposed relation by identifying the salient discourse cues, rather than simply paraphrasing or restating the dialogue.

To assess rationale quality, two authors independently evaluated a random sample of 50 relation rationales using a 1–5 Likert scale. The rationales achieved high average scores on both dimensions, with a mean faithfulness score of 4.88 and a mean helpfulness score of 4.72. Inter-annotator agreement was also high, with identical ratings assigned to 47/50 rationales (94%) for faithfulness and 42/50 rationales (84%) for helpfulness.