Improving Dialogue Discourse Parsing through Discourse-aware Utterance Clarification

Yaxin Fan, Peifeng Li*, and **Qiaoming Zhu**
School of Computer Science and Technology, Soochow University, Suzhou, China
yxfansuda@stu.suda.edu.cn, {pfli, qmzhu}@suda.edu.cn

Abstract

Dialogue discourse parsing aims to identify and analyze discourse relations between the utterances within dialogues. However, linguistic features in dialogues, such as omission and idiom, frequently introduce ambiguities that obscure the intended discourse relations, posing significant challenges for parsers. To address this issue, we propose a Discourse-aware Clarification Module (DCM) to enhance the performance of the dialogue discourse parser. DCM employs two distinct reasoning processes: clarification type reasoning and discourse goal reasoning. The former analyzes linguistic features, while the latter distinguishes the intended relation from the ambiguous one. Furthermore, we introduce Contribution-aware Preference Optimization (CPO) to mitigate the risk of erroneous clarifications, thereby reducing cascading errors. CPO enables the parser to assess the contributions of the clarifications from DCM and provide feedback to optimize the DCM, enhancing its adaptability and alignment with the parser's requirements. Extensive experiments on the STAC and Molweni datasets demonstrate that our approach effectively resolves ambiguities and significantly outperforms the state-of-the-art (SOTA) baselines.[1]

1 Introduction

Dialogue discourse parsing focuses on uncovering the implicit discourse structure within dialogues by constructing a relation-based dependency tree. Understanding the discourse structure is advantageous for various downstream tasks, including dialogue generation (Li et al., 2024c; Fan et al., 2024b), meeting summarization (Gao et al., 2023), sentiment analysis (Li et al., 2023c), and reading comprehension (Li et al., 2023e). Figure 1 illustrates an example from the STAC dataset (Asher et al., 2016), comprising six utterances (u_1–u_6) from three speakers. A dialogue discourse parser aims to predict dependent utterances for each utterance in a dialogue and identify their corresponding relation types.

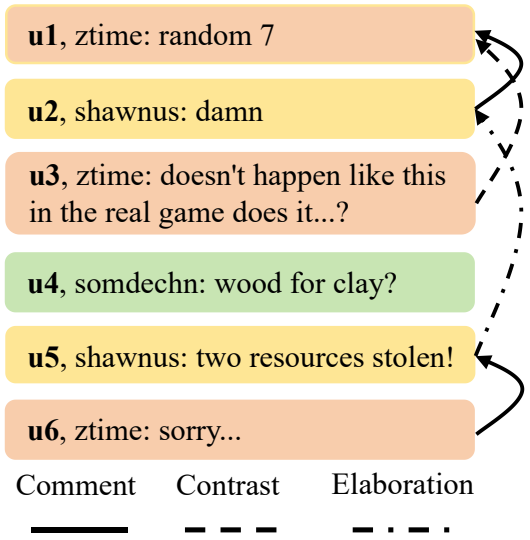

Figure 1: An example from STAC (Asher et al., 2016) dataset. The utterance u_4 has no dependent utterance.

Recent advancements have utilized the robust contextual understanding of open-source Large Language Models (LLMs) (Touvron et al., 2023; Grattafiori et al., 2024) to improve discourse parsing from both input and output perspectives. These advancements include (1) integrating historical structures into the input (Thompson et al., 2024), (2) generating sophisticated output that aligns with natural language (Li et al., 2024a), and (3) providing detailed explanations of discourse relations in the output (Liu et al., 2025).

Despite significant advancements, existing studies primarily focus on adapting LLMs for discourse parsing, often overlooking challenges posed by the intrinsic linguistic features of dialogues. These features introduce ambiguities that can significantly impair the performance of discourse parsers. For example, the utterance u_6 in Figure 1 exempli-

*Corresponding author
[1]We released our code at https://github.com/yxfanSuda/DCM

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 18800–18816
July 27 - August 1, 2025

fies the ambiguity caused by omissions. Since u_6 merely contains "sorry" without specifying its referent, it is challenging to determine whether the apology pertains to the resource theft mentioned in u_5 or the rejection of the resource exchange request in u_4. Additional examples of linguistic features that lead to ambiguity, such as typos, abbreviations, slang, and idioms are provided in Appendix A.

To address these challenges, we present a Discourse-aware Clarification Module (DCM), designed to provide clarifications for the parser, thereby reducing ambiguity in conversational understanding. DCM employs two key reasoning mechanisms: clarification type reasoning and discourse goal reasoning. The former provides directives for generating clarifications, such as adding omitted content or correcting typographical errors, while the latter guides the clarification to align more closely with the intended discourse relation by contrasting it with the ambiguous one. For instance, as illustrated in Figure 1, clarification type reasoning first identifies the omission in u_6. Following this, discourse goal reasoning ensures that the added content clarifies u_6 as an apology directed at u_5, rather than a refusal of u_4.

To further minimize erroneous clarifications, we propose a Contribution-aware Preference Optimization (CPO) to mitigate the risk of erroneous clarifications in DCM. CPO enables the parser to assess the contributions of the clarifications from DCM and provide feedback to optimize DCM, enhancing its adaptability and alignment with the parser's requirements.

We validate the effectiveness of our approach through extensive experiments on two widely used dialogue discourse datasets, STAC and Molweni. The results demonstrate that our approach significantly outperforms the state-of-the-art (SOTA) baselines. An in-depth analysis shows that our DCM effectively eliminates ambiguity through discourse-aware clarification, while CPO further reduces the introduction of erroneous clarifications, leading to more robust parsing performance.

2 Related Work

Previous research on dialogue discourse parsing predominantly falls into two categories: discriminative and generative approaches. Discriminative methods typically predict discourse links and relations by calculating the probabilities between utterances. These studies often enhance parsing performance by modeling key elements within dialogues, such as speakers (Ji and Kong, 2023; Jiang et al., 2023; Yu et al., 2022), utterances (Mao et al., 2023; He et al., 2021; Yang et al., 2021), and dialogue structure (Wang et al., 2024; Fan et al., 2023; Li et al., 2023d; Chi and Rudnicky, 2022; Fan et al., 2022; Yang et al., 2021; Wang et al., 2021; Shi and Huang, 2019). Furthermore, some research has addressed data sparsity issues by exploring cross-domain (Liu and Chen, 2021), semi-supervised (Li et al., 2023a), and unsupervised methods (Cimino et al., 2024; Li et al., 2024a, 2023b).

On the other hand, generative methods utilize generative models to generate discourse links and relations, often represented through natural language descriptions. Wang et al. (2023a) pioneered the application of the generative paradigm in dialogue discourse parsing, achieving significant success. However, Chan et al. (2024) and Fan et al. (2024a) assessed ChatGPT's performance, finding it still falls short of the SOTA models. In response, some research has focused on fine-tuning open-source LLMs for dialogue discourse parsing. Li et al. (2024b) explored advanced representations to enhance the naturalness of outputs. Moreover, Thompson et al. (2024) introduced an incremental discourse parser by integrating historical structures, while Liu et al. (2025) enhanced discourse parsing through explanation generation.

Despite these advancements, previous research has often overlooked the inherent linguistic features in dialogues that introduce ambiguity, posing challenges to discourse parsing. Therefore, we introduce an innovative discourse-aware clarification module that clarifies utterances to eliminate ambiguity, thereby enhancing discourse parsing.

3 Preliminaries

Given a discourse training set $\mathcal{D} = \{(d^i, y^i)\}_{i=1}^N$, each instance consists of:

- A dialogue history $d^i = \{(s_t^i, u_t^i)\}_{t=1}^k$ comprising k turns, where s_t^i and u_t^i represent the speaker and utterance at the turn t, respectively.
- A discourse relation, represented as a triplet $y^i = (k, t', r)$, indicates that the current utterance u_k^i in d^i is connected to its dependent utterance $u_{t'}^i$ $(1 \leq t' < k)$ via a relation $r \in \mathcal{R}$.

For each d^i, the discourse parser, an autoregressive model, $\mathcal{DP}$ first identifies a link between the current utterance u_k^i and the utterance $u_{t'}^i$ $(1 \leq t' <$

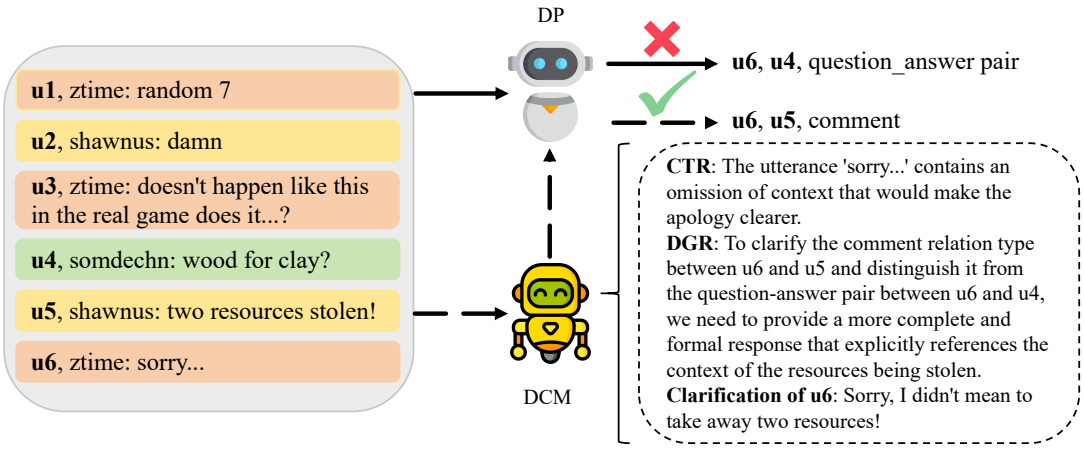

Figure 2: Framework of our method, where u_6 is the utterance being parsed. Solid lines represent direct parsing, while dashed lines indicate the enhancement of discourse parsing through a discourse-aware clarification module.

k) in the history d^i, and then generates their relation type r (e.g., Comment), while DCM ($\mathcal{DCM}$) is to replace the current utterance u_k^i with a clarified utterance u_c^i. They can be formally as follows:

$$\begin{aligned} u_c^i &\leftarrow \mathcal{DCM}(d^i, u_k^i), \\ y^i &\leftarrow \mathcal{DP}\left(d^i, u_c^i\right). \end{aligned} \tag{1}$$

4 Methodology

Our approach is illustrated in Figure 2. The Discourse-aware Clarification Module (DCM) enhances the performance of the Discourse Parser (DP) by providing clarifications through Clarification Type Reasoning (CTR) and Discourse Goal Reasoning (DGR).

4.1 Discourse Parser

Following prior work (Thompson et al., 2024; Liu et al., 2025), we fine-tune an open-source LLM to function as a discourse parser. The input and output formats are detailed in Appendix B.1. For each input-output pair (d^i, y^i) in $\mathcal{D}$, the parser is trained to generate y^i conditioned on d^i by minimizing the negative log-likelihood:

$$\mathcal{L}_\theta = -\frac{1}{N}\sum_{i=1}^{N} \log p_\theta(y^i \mid d^i), \tag{2}$$

where θ represents the trainable parameters of the parser, and $p_\theta(y^i \mid d^i)$ denotes the probability distribution over the generation of y^i given the input d^i.

4.2 Discourse-aware Clarification Module

DCM must be customized to address the parser's specific requirements, focusing on two critical issues. The first issue is the identification of clarification types. Given the diverse linguistic features in dialogues, such as omissions and typos that can cause ambiguity, it is essential for DCM to accurately identify these types to provide appropriate clarifications, such as supplementing omissions or correcting typos. The second issue pertains to resolving the parser's ambiguous discourse relations. DCM must ensure that clarifications align with the intended discourse relations, thereby eliminating ambiguity in the parser's understanding. To address these challenges, we designed clarification type reasoning CTR and discourse goal reasoning DGR to guide the generation of clarifications.

Figure 3 illustrates its training process. Initially, we utilize automatically generated clarification data for supervised fine-tuning, thereby endowing DCM with discourse-aware clarification capabilities. Subsequently, we introduce a Contribution-aware Preference Optimization (CPO), which minimizes the erroneous clarifications by DCM and enhances its adaptability to the parser.

Clarification Type Reasoning CTR analyzes linguistic features in utterances that may cause ambiguity, providing the directive for clarification. As illustrated in Figure 2, CTR identifies omissions in the utterance u_6, suggesting that addressing these omissions would make the apology clearer.

To meet the specific requirements of the parser, CTR is designed to systematically address linguistic features that frequently induce parsing errors. Through a systematic error analysis (see Appendix C), our investigation identified five key linguistic features that could lead to ambiguity in dialogue understanding by the parser: omission, ty-

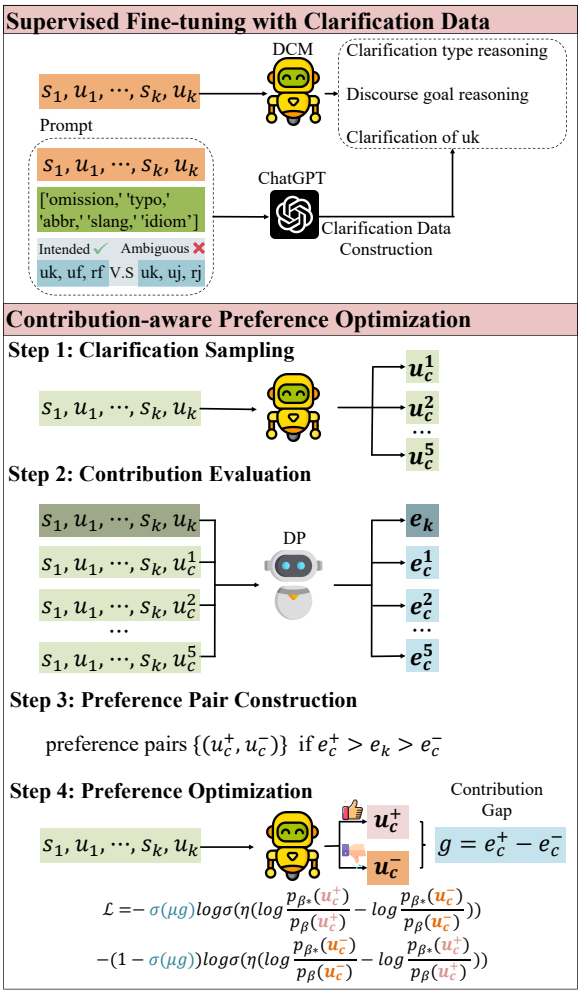

Figure 3: Training process of DCM.

pos, abbreviations, slang, and idioms. CTR detects these five linguistic features in the utterance and guides the generation of appropriate clarifications.

Discourse Goal Reasoning DGR emphasizes the importance of aligning clarifications with the intended discourse relation. It achieves this by contrasting the intended relation with the ambiguous one identified by the parser. As illustrated in Figure 2, DGR highlights the intended comment relation type between u_6 and u_5. Furthermore, it distinguishes this intended relation from the ambiguous question-answer pair between u_6 and u_4. DGR suggests explicitly referencing the context of the "stolen resources" mentioned in u_5 within the clarifications.

Guided by clarification type reasoning and discourse goal reasoning, DCM generates clarifications for the parser, effectively eliminating ambiguity and improving parsing performance.

Implementation To obtain training data incorporating clarification type reasoning, discourse goal reasoning, and the final clarification for fine-tuning DCM, we follow previous work (Liu et al., 2025; Wang et al., 2023b) by leveraging ChatGPT to automatically generate the data. We randomly selected $\alpha\%$ of the data from $\mathcal{D}$ as the seed dataset, denoted as $\mathcal{D}^T$, with the number of instances denoted as M. The construction process is illustrated in Figure 3. A tailored prompt (see Appendix B.2) was designed to guide ChatGPT in generating the clarification data. This prompt comprises four key elements: dialogue history, a list of clarification types, intended discourse relation, and ambiguous discourse relation.

To derive the ambiguous discourse relation y^i_{am}, we trained a discourse parser on the remaining $(1-\alpha)\%$ of $\mathcal{D}$ and tested it on $\mathcal{D}^T$. Let $\hat{y}^i$ be the parser's prediction for dialogue d^i in $\mathcal{D}^T$, the ambiguous discourse relation y^i_{am} is defined as:

$$y^i_{\text{am}} = \begin{cases} \hat{y}^i, & \text{if } \hat{y}^i \neq y^i \\ y^i_{\text{ps}}, & \text{if } \hat{y}^i = y^i \end{cases} \quad (3)$$

Here, y^i_{ps} represents the pseudo-ambiguous relation that we construct for the samples correctly predicted by the discourse parser. In these cases, while the dependent utterance remains unchanged, the relation type is randomly altered. This improves the DCM's adaptability in handling utterances that are already correctly understood by the discourse parser.

Guided by the prompt, ChatGPT executes a sequential process involving clarification type reasoning, and discourse goal reasoning, and concludes with the clarification. An example is shown in Figure 2 (lower right).

Finally, we fine-tune an open-source LLM as DCM. The input and output formats are detailed in Appendix B.3. Let t represent the text containing CTR, DGR, and the final clarification u_c, DCM is trained to generate t^i conditioned on d^i in $\mathcal{D}^T$ by minimizing the negative log-likelihood:

$$\mathcal{L}_\beta = -\frac{1}{M}\sum_{i=1}^{M} \log p_\beta(t^i \mid d^i) \quad (4)$$

where β is the parameter and p_β indicates the probabilities that generate the t^i given the input d^i.

4.3 Contribution-aware Preference Optimization

Since the DCM is trained on automatically constructed clarification data, it may inadvertently introduce erroneous clarifications into the parser. To

address this issue, we propose Contribution-aware Preference Optimization (CPO), a method in which the parser evaluates the contributions of clarifications generated by the DCM and provides feedback to guide the DCM's optimization, thereby reducing cascading errors. As shown in Figure 3, CPO consists of four steps: clarification sampling, contribution evaluation, preference pair construction, and preference optimization.

Clarification Sampling To construct preference data, we use the remaining $(1-\alpha)\%$ of $\mathcal{D}$, denoted as $\mathcal{D}^C$, as the seed dataset. We employ the fine-tuned DCM to sample 5 clarified utterances $\{u_c^j\}_{j=1}^5$ for each dialogue history d in $\mathcal{D}^C$.[2] This self-sampling strategy not only provides diverse candidate clarifications for preference optimization but also reduces excessive reliance on prompting closed-source LLMs.

Contribution Evaluation We employ the fine-tuned parser to evaluate the contribution of each clarification u_c^j to accurate parsing. Specifically, the parser calculates the log probability e_c^j of generating the intended discourse relations y, conditioned on the clarified utterance u_c^j as follows:

$$e_c^j = logp_\theta(y \mid s_1, u_1, \cdots, s_k, u_c^j) \tag{5}$$

Here, e_c^j represents the contribution score of u_c^j, where higher scores indicate a greater likelihood of DP achieving the intended relation.

Preference Pair Construction For each example in $\mathcal{D}^C$, we construct the pairwise preference data $\{(d, u_c^+, u_c^-, g)\}$, where u_c^+ and u_c^- are the concatenation of the clarified utterances in $\{u_c^j\}_{j=1}^5$. Specifically, u_c^+ is chosen from $\{u_c^j \mid e_c^j > e_k\}$, and u_c^- is chosen from $\{u_c^j \mid e_c^j < e_k\}$. Here, e_k represents the log probability of generating the intended discourse relation y conditioned on the original utterance u_k. [3] Let $e_c^{+/-}$ denote the contribution score of $u_c^{+/-}$, by setting $e_c^+ > e_k$ and $e_c^- < e_k$, we ensure that u_c^+ is preferred over u_c^-, as u_c^+ demonstrates a higher likelihood of enabling the discourse parser to correctly predict y compared to the original utterance u_k. The term $g = e_c^+ - e_c^-$ quantifies the contribution gap between the preference pair, reflecting how much more u_c^+ contributes to the correct prediction of y compared to u_c^-.

[2]We experimented with different sampling frequencies (3, 5, and 10) and found that sampling 5 times yielded the best performance on the validation set.

[3]Examples were discarded if all clarified utterances belonged exclusively to either $\{u_c^j \mid e_c^j > e_k\}$ or $\{u_c^j \mid e_c^j < e_k\}$.

Preference Optimization Direct Preference Optimization (DPO) (Rafailov et al., 2023) has been widely used to align LLMs with human preferences by maximizing the contrast between preferred and non-preferred candidates. However, DPO treats each preference pair equally, which could lead to excessive optimization of minor differences when the contribution gap is small, making the model prone to overfitting. To address this issue, we assign different weights to preference pairs, giving more attention to those with larger contribution gaps and less attention to those with smaller gaps. The contribution gap g is incorporated into the DPO loss, and the training objective is as follows:

$$\begin{aligned}\mathcal{L}_{cpo} = -\frac{1}{N'}\sum_{i=1}^{N'}[\sigma(\mu g^i)\log \sigma f(u_c^{i+}, u_c^{i-}, d^i) \\ +(1-\sigma(\mu g^i))\log \sigma f(u_c^{i-}, u_c^{i+}, d^i)]\end{aligned} \tag{6}$$

$$f(u^+, u^-, d) = \eta(\log \frac{p_{\beta^*}(u^+|d)}{p_\beta(u^+|d)} - \log \frac{p_{\beta^*}(u^-|d)}{p_\beta(u^-|d)}) \tag{7}$$

Here, N' is the number of preference pairs, β^* represents the trainable parameters of DCM in preference optimization, while β denotes the frozen parameters of the DCM after supervised fine-tuning. The function σ is the sigmoid function, η is a hyperparameter of DPO, and μ is a scaling factor to smooth the training process. When g^i is large, $\sigma(\mu g^i)$ approaches 1, thereby drawing significant attention to the preference pair. Conversely, when g^i is small, $\sigma(\mu g^i)$ approaches 0.5, resulting in minimal attention to the preference pair. Notably, $\mathcal{L}_{cpo}$ simplifies to the standard DPO loss when $\sigma(\mu g^i)$ equals 1.

4.4 Training and Inference

We minimize the losses $\mathcal{L}_\theta$ and $\mathcal{L}_\beta$ to fine-tune DP and DCM, respectively. For preference optimization, we minimize the loss $\mathcal{L}_{cpo}$ to improve DCM's adaptability to the parser.

In the inference stage, only those samples where the parser exhibits uncertainty are processed by DCM for clarification. To assess uncertainty, we employ a self-sampling method. For a given test sample d, it is first processed by the parser to generate o times predictions, denoted as $\{\hat{y}_j\}_{j=1}^o$. A majority voting mechanism is then applied to determine the final prediction $\hat{y}$. If $\hat{y}$ appears more than $o/2$ times, it indicates that the parser has strong confidence, and $\hat{y}$ is accepted as the final prediction. Otherwise, this test sample is forwarded to

DCM for clarification, after which it is processed again by the parser, and the final prediction is determined again through majority voting. This ensures that only ambiguous cases undergo additional clarification, avoiding unnecessary clarifications.

5 Experimentation

5.1 Experimental Setup

Datasets We conducted experiments on two widely used dialogue discourse datasets: STAC (Asher et al., 2016) and Molweni (Li et al., 2020). The STAC dataset, a multi-party dialogue corpus derived from an online game, comprises 1,062 dialogues for training and 111 dialogues for testing. These sets respectively include 11,703 and 1,132 discourse relations. In line with prior research, we randomly selected 10% of the training dialogues for validation purposes. The Molweni dataset, derived from the Ubuntu Chat Corpus (Lowe et al., 2015), is structured into 9,000 dialogues for training, 500 for validation, and 500 for testing. This distribution encompasses 70,454, 3,880, and 3,911 discourse relations in the training, validation, and testing sets, respectively. Both datasets define 16 distinct relation types: comment, clarification-question, elaboration, acknowledgment, continuation, explanation, conditional, question-answer pair, alternation, question-elaboration, result, background, narration, correction, parallel, and contrast.

Evaluation Metric Following previous work (Liu et al., 2025; Thompson et al., 2024), we adopted micro-averaged F_1 for both link prediction (L F_1) and link&relation prediction (LR F_1). L F_1 measures the performance of correct link prediction (Link or Non-link), while LR F_1 evaluates the performance of simultaneous prediction of both the link and the relation type.

Implementation Details Following previous work (Thompson et al., 2024; Liu et al., 2025), we used the widely adopted open-source LLM, LLaMA3[4](Grattafiori et al., 2024) as the backbone for our experiments. We adopted LoRA (Hu et al., 2022) for parameter-efficient fine-tuning of LLaMA3, setting the rank and scaling parameters to 8 and 16, respectively. For constructing clarification data, we used the GPT-4 (version: 2024-08-06) model. During fine-tuning, both DP and DCM used the same backbone, with inputs limited to the 20 most recent utterances. The parameter α, which controls the proportion of data used for supervised fine-tuning of DCM, was set to 10% and 20% for STAC and Molweni, respectively. Further analysis of α is provided in Appendix D. The training hyperparameters for DP and DCM are listed in Appendix E. During self-sampling for DP and DCM, the hyperparameters temperature, top_p, and max_output_length were set to 0.6, 0.9, and 512, respectively. The number of prediction trials o was set to 10. The best model was selected based on validation set performance. All experiments were conducted using the LLaMA-Factory[5] (Zheng et al., 2024) framework on two RTX 4090D GPUs.

5.2 Baselines

We compare our method against both discriminative and generative baselines.

Discriminative Methods **SSAM** (Wang et al., 2021): It captures global dialogue structure using a graph transformer, introducing two auxiliary training signals for enhanced discourse parsing. **SSP** (Yu et al., 2022): It enhances speaker interaction through a second-stage pre-training task. **DAMT** (Fan et al., 2022): It fuses results from various decoding paradigms to improve discourse parsing. **SDDP** (Chi and Rudnicky, 2022): It uses structured encoding of the adjacency matrix to jointly optimize discourse links and relations. **DialogDP** (Li et al., 2023d): It combines top-down and bottom-up parsing strategies. **RLTST** (Fan et al., 2023): It leverages reply-to structures for addressee recognition to aid discourse parsing. **UniMPC** (Xie et al., 2024): It proposes a unified framework to consolidate common sub-tasks in multi-party dialogue understanding.

Generative Methods **ChatGPT** (Fan et al., 2024a): It directly evaluates ChatGPT's performance in discourse parsing. **D^{2}PSG** (Wang et al., 2023a): It introduces the generative paradigm to discourse parsing, exploring model comprehension of discourse relations. **Seq2Seq-DDP** (Li et al., 2024b): It develops advanced representations to align outputs more closely with natural language. **Llampia** (Thompson et al., 2024): It proposes an incremental discourse parser by incorporating predicted historical structures. **DDPE** (Liu et al., 2025): It enhances discourse parsing through explanation generation.

[4] https://huggingface.co/meta-llama/Meta-Llama-3-8B

[5] https://github.com/hiyouga/LLaMA-Factory

Type	Model	LLM	STAC		Molweni	
			L F_1	LR F_1	L F_1	LR F_1
Discriminative	SSAM	ELECTRA-small	73.5	57.3	81.6	58.5
	SSP	BERT-base	73.0	57.4	83.7	59.4
	DAMT	XLNet-base	73.6	57.4	82.5	58.9
	SDDP	RoBERTa-base	74.4	59.6	83.5	59.9
	DialogDP	BERT-large	73.0	58.5	83.2	59.8
	RLTST	BERT-base	73.7	57.6	85.3	60.9
	UniMPC	RoBERTa-base	72.8	56.7	79.6	57.3
Generative	ChatGPT	-	59.9	25.3	63.8	23.9
	Seq2Seq-DDP‡	T0 (3B)	72.3	56.6	83.4	60.0
	D^2PSG‡	T5-large (0.8B)	78.4	62.8	87.1	62.0
	Llampia†	LLaMA3 (8B)	77.5	60.7	-	-
	DDPE † (SOTA)	LLaMA3 (8B)	79.5	63.4	87.6	62.9
	DP-DCM-CPO†(Ours)	LLaMA3 (8B)	**82.2**	**69.0**	**88.5**	**66.2**
	w/o CPO		79.9	65.5	87.6	63.8
	w/o DCM&CPO		77.8	63.2	86.8	62.3

Table 1: Experimental results on STAC and Molweni, where ‡ denotes full fine-tuning, while † represents parameter-efficient fine-tuning with LoRA. The performance improvement of our DP-DCM-CPO over the SOTA DDPE is statistically significant, as confirmed by a t-test with a p-value < 0.05.

5.3 Overall Performance

Table 1 presents the experimental results of our DP-DCM-CPO on both the STAC and Molweni datasets. The results demonstrate that our method achieves SOTA performance, surpassing both discriminative and generative baselines by substantial margins. The results show that most generative methods outperform discriminative methods, especially fine-tuned LLM-based approaches. Compared with the SOTA generative method DDPE, our method exhibits 2.7/5.6-point advantages in L F_1/LR F_1 on STAC and 0.9/3.3-point improvements on Molweni. These results strongly validate the effectiveness and generalization capability of our method.

In addition, although both Llampia and DDPE employ parameter-efficient fine-tuning on the 8B-parameter LLaMA3, their performance only matches that of the fully fine-tuned D^2PSG (0.8B parameters). In contrast, our method achieves superior performance by eliminating ambiguity through discourse-aware clarification. Additional experimental results with different backbones and parameter sizes are provided in Appendix F.

Notably, the improvement in the LR F_1 metric is significantly more pronounced than that in the L F_1 metric. This can be attributed to two factors: (1) the L F_1 metric itself has already reached a high-performance level, and (2) our method effectively mitigates the issue of relation confusion.

Furthermore, we observe that the performance gains on Molweni are less substantial compared to those on STAC. This discrepancy is likely due to the inherent differences between the two corpora: STAC, derived from an online game, contains diverse expressions and linguistic features as discussed in the Introduction, whereas Molweni, sourced from Ubuntu technical discussions, features highly technical dialogues with fewer informal expressions and low-frequency linguistic features.

6 Analysis

6.1 Analysis of DCM

We conducted ablation experiments to evaluate the effectiveness of DCM, as summarized in Table 1 (w/o DCM&CPO). Since CPO is designed to enhance DCM, removing DCM also necessitates the removal of CPO. The ablation results demonstrate that the removal of DCM leads to a performance degradation of 4.4 points in L F_1 and 5.8 points in LR F_1 on STAC. Similarly, on the Molweni dataset, L F_1 and LR F_1 decrease by 1.7 and 3.9 points, respectively. These findings highlight the significant role of DCM in improving the parser's performance.

To further investigate the impact of individual components within DCM, we analyzed the per-

Category	Omission	Typo	Others
Percentage(%)	60	25	15
Accuracy(%)			
DCM	34.5	9.3	11.4
w/o CTR	33.3	9.3	2.8
w/o DGR	28.5	8.3	8.5

Table 2: Performance degradation on STAC across different types caused by the removal of CTR and DGR. The category "Others" includes abbreviation, slang, and idiom.

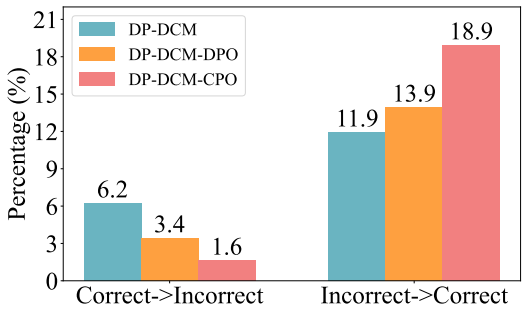

Figure 4: Comparison of our CPO with standard DPO on STAC in two scenarios.

formance degradation on STAC across different clarification types caused by the removal of CTR and DGR, as illustrated in Table 2. We observed that our DCM primarily addresses omission, which constitutes the largest proportion of errors in the dataset. When CTR is removed, DCM struggles to process the 'Others' category involving abbreviations, slang, and idioms. Such expressions often extend beyond the immediate dialogue context, making accurate interpretation more challenging. This suggests that CTR enhances DCM's ability to capture implicit meanings in metaphorical or culturally specific contexts. It achieves this by explicitly analyzing clarification types, thereby generating more contextually relevant clarifications.

Moreover, the absence of DGR results in the most pronounced decline in addressing omissions. This indicates that DGR plays a pivotal role in enabling DCM to infer the underlying intent behind such omissions by contrasting intended and ambiguous discourse relations. In doing so, DGR helps DCM generate clarified utterances that better reflect the intended discourse relations, ultimately enhancing parsing performance. Similar trends are observed in Molweni, as illustrated in Appendix G.1. These findings highlight the complementary roles of CTR and DGR in enhancing DCM's effectiveness and robustness.

6.2 Analysis of CPO

To analyze the effectiveness of the proposed CPO, we conducted ablation experiments as shown in Table 1 (w/o CPO). The ablation results demonstrate that the removal of CPO leads to a performance degradation of 2.3 points in L F_1 and 3.5 points in LR F_1 on STAC. Similarly, on Molweni, L F_1 and LR F_1 decrease by 0.9 and 2.4 points, respectively. These results demonstrate the effectiveness of our CPO method.

Furthermore, we analyzed the distribution of two scenarios following DCM's clarifications to DP: 1) Correct->Incorrect: DP initially predicted correctly but predicted incorrectly after DCM clarification. 2) Incorrect->Correct: DP initially predicted incorrectly but predicted correctly after DCM clarification. The distribution in STAC is illustrated in Figure 4. DP-DCM-DPO, which employs the standard DPO, differs from CPO by setting $\sigma(\mu g_i)$ in Equation 6 to 1.

We observed that 6.2% of DP's initially correct predictions became incorrect after DCM clarification when CPO was removed. This may be due to the unavoidable noise introduced by the automatically constructed data used to train DCM, as further discussed in Section 6.3. By enhancing the adaptability of DCM to DP with DPO, the Correct->Incorrect proportion is effectively reduced to 3.4%. Notably, our CPO enhances DCM by capturing the contribution gaps of preference pairs, reducing the proportion of Correct->Incorrect to 1.6%. Furthermore, CPO significantly increases the proportion of Incorrect->Correct from 11.9% to 18.9%, compared to DPO's 13.9%. The distribution in Molweni (in Appendix G.2) shows a similar pattern to that of STAC. These results demonstrate the effectiveness of our CPO in reducing erroneous clarifications by DCM, thereby enhancing parsing performance.

6.3 Quality Analysis of Clarification Data

To evaluate the quality of the clarification data, we conducted a manual pairwise evaluation to assess whether the clarified or original utterances more clearly conveyed the intended relation types with their dependent utterances. Further details are provided in Appendix G.3.

Figure 5 presents the evaluation results of the clarifications generated by ChatGPT, DCM, and DCM-CPO. The terms "Win," "Tie," and "Lose"

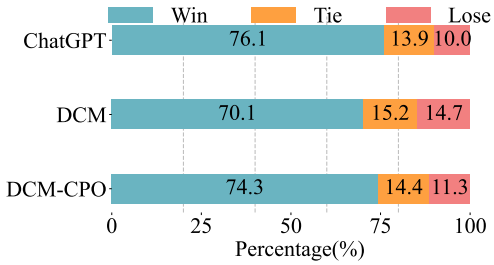

Figure 5: Results of the human pairwise evaluation of clarification data quality on STAC.

denote cases where the clarified utterance is superior to, equivalent to, or inferior to the original utterance. As shown in the figure, 76.1% of the clarified utterances generated by ChatGPT were superior, while 10.0% were inferior to the original utterances. This demonstrates that most data constructed using ChatGPT is satisfactory, some degree of noise is inevitable. Our DCM, trained on data from ChatGPT, generated 70.1% superior and 14.7% inferior clarifications, indicating that noise inevitably affects its performance. However, our CPO, which optimizes DCM by leveraging parser feedback, mitigates the effects of noise, resulting in 74.3% superior and 11.3% inferior clarifications compared to the original utterances. Similar patterns are observed in the Molweni dataset, as detailed in Appendix G.3.

These findings suggest that while using ChatGPT to automatically construct data is efficient and cost-effective, it introduces noise that affects DCM's performance. Although our CPO mitigates the impact of this noise, it cannot eliminate it entirely. Future work should focus on enhancing clarification quality to further advance discourse parsing.

6.4 Performance Using Open-source Clarification Data

To demonstrate the generalizability of our method, we conducted experiments using clarification data generated by two popular open-source LLMs: Vicuna-13B-v1.3 [6] and DeepSeek-V3 [7]. As shown in Table 3, even when trained with data from these open-source LLMs, the discourse-aware clarification module can markedly enhance the performance of the discourse parser. This result indicates that our method does not depend on the superior performance of closed-source LLMs, thereby demonstrating increased robustness and generalizability.

[6] https://huggingface.co/lmsys/vicuna-13b-v1.3

[7] https://huggingface.co/deepseek-ai/DeepSeek-V3

Model	Data	STAC		Molweni	
		L F_1	LR F_1	L F_1	LR F_1
DP	-	77.8	63.2	86.8	62.3
Ours	Vicuna	80.7	66.2	87.8	63.3
	DeepSeek	81.2	68.0	88.0	65.6

Table 3: Experimental results utilizing open-source clarification data. The backbone model for DP and our method is LLaMA3-8b.

6.5 Case Study

We conducted case studies to further demonstrate the effectiveness of our method. Figure 2 presents an example of ambiguity caused by omission. The lack of referential content in utterance u_6 led the parser to incorrectly parse the relation type between u_6 and u_4 as a question_answer pair. Our DCM clarifies utterance u_6, by adding the necessary referential content, which enables DP to correctly identify the comment relation type between u_6 and u_5. Other types of examples can be found in Appendix G.4.

6.6 Analysis of Uncertainty Assessment

In Appendix G.5, we examine the impact of our uncertainty assessment method during the inference stage. The findings demonstrate that our method effectively distinguishes between uncertain and certain instances, enabling targeted improvements in overall parsing performance.

7 Conclusion

In this paper, we introduce a Discourse-aware Clarification Module (DCM) aimed at reducing ambiguity in dialogue parsing. DCM generates clarifications for the parser through systematic clarification type reasoning and discourse goal reasoning. Additionally, we propose the Contribution-aware Preference Optimization (CPO) method, which optimizes DCM based on feedback from the parser, thereby reducing erroneous clarifications by DCM. Extensive experiments on the STAC and Molweni datasets demonstrate the effectiveness of our approach. Future work will focus on enhancing the quality of clarification data to further enhance discourse parsing.

Limitations

Our primary limitation lies in the quality of automatically constructed clarification data. While employing closed-source or open-source LLMs to generate the data saves time and costs, the quality and consistency of the generated data can vary. LLMs, such as ChatGPT, occasionally generate irrelevant or contextually inappropriate responses. This inconsistency can undermine the reliability of the clarification data, posing challenges for the adaptability of our discourse-aware clarification module to discourse parsers. While our proposed CPO method can mitigate the impact of noise to some extent, it cannot completely eliminate it. Future work needs to focus more on obtaining high-quality clarification data to further enhance the overall performance of the discourse parser.

Acknowledgements

The authors would like to thank the three anonymous reviewers for their comments on this paper. This research was supported by the National Natural Science Foundation of China (Nos. 62276177 and 62376181), and Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

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 LREC*, pages 2721–2727.

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

Chunkit Chan, Cheng Jiayang, Weiqi Wang, Yuxin Jiang, Tianqing Fang, Xin Liu, and Yangqiu Song. 2024. Exploring the potential of ChatGPT on sentence level relations: A focus on temporal, causal, and discourse relations. In *Findings of the EACL*, pages 684–721.

Ta-Chung Chi and Alexander Rudnicky. 2022. Structured dialogue discourse parsing. In *Proceedings of the SIGDIAL*, pages 325–335.

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 SIGDIAL*, pages 297–316.

Yaxin Fan, Feng Jiang, Peifeng Li, Fang Kong, and Qiaoming Zhu. 2023. Improving dialogue discourse parsing via reply-to structures of addressee recognition. In *Proceedings of the EMNLP*, pages 8484–8495.

Yaxin Fan, Feng Jiang, Peifeng Li, and Haizhou Li. 2024a. Uncovering the potential of ChatGPT for discourse analysis in dialogue: An empirical study. In *Proceedings of the LREC-COLING*, pages 16998–17010.

Yaxin Fan, Peifeng Li, Fang Kong, and Qiaoming Zhu. 2022. A distance-aware multi-task framework for conversational discourse parsing. In *Proceedings of the COLING*, pages 912–921.

Yaxin Fan, Peifeng Li, and Qiaoming Zhu. 2024b. Improving multi-party dialogue generation via topic and rhetorical coherence. In *Proceedings of the EMNLP*, pages 3240–3253.

Shen Gao, Xin Cheng, Mingzhe Li, Xiuying Chen, Jinpeng Li, Dongyan Zhao, and Rui Yan. 2023. Dialogue summarization with static-dynamic structure fusion graph. In *Proceedings of the ACL*, pages 13858–13873.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yuchen He, Zhuosheng Zhang, and Hai Zhao. 2021. Multi-tasking dialogue comprehension with discourse parsing. In *Proceedings of the PACLIC*, pages 598–608.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of the ICLR*, pages 1–13.

Shaoming Ji and Fang Kong. 2023. Speaker-aware dialogue discourse parsing with meta-path based heterogeneous graph neural network. In *Proceedings of the ICIC*, pages 575–586.

Yuru Jiang, Yu Li, Weikai He, Jie Chen, Yanchao Yu, and Yangsen Zhang. 2023. A new dataset and parsing model for chinese multiparty dialogue discourse structure. In *Proceedings of the IALP*, pages 221–227.

Chuyuan Li, Maxime Amblard, and Chloé Braud. 2023a. A semi-supervised dialogue discourse parsing pipeline. In *Proceedings of the LIFT*, pages 1–10.

Chuyuan Li, Chloé Braud, Maxime Amblard, and Giuseppe Carenini. 2024a. Discourse relation prediction and discourse parsing in dialogues with minimal supervision. In *Proceedings of the CODI*, pages 161–176.

Chuyuan Li, Patrick Huber, Wen Xiao, Maxime Amblard, Chloe Braud, and Giuseppe Carenini. 2023b. Discourse structure extraction from pre-trained and fine-tuned language models in dialogues. In *Findings of the EACL*, pages 2562–2579.

Chuyuan Li, Yuwei Yin, and Giuseppe Carenini. 2024b. Dialogue discourse parsing as generation: A sequence-to-sequence LLM-based approach. In *Proceedings of the SIGDIAL*, pages 1–14.

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 COLING*, pages 2642–2652.

Jingyang Li, Shengli Song, Yixin Li, Hanxiao Zhang, and Guangneng Hu. 2024c. ChatMDG: A discourse parsing graph fusion based approach for multi-party dialogue generation. *Information Fusion*, 110:102469.

Wei Li, Luyao Zhu, Rui Mao, and Erik Cambria. 2023c. SKIER: A symbolic knowledge integrated model for conversational emotion recognition. In *Proceedings of the AAAI*, pages 13121–13129.

Wei Li, Luyao Zhu, Wei Shao, Zonglin Yang, and Erik Cambria. 2023d. Task-aware self-supervised framework for dialogue discourse parsing. In *Findings of the EMNLP*, pages 14162–14173.

Yanling Li, Bowei Zou, Yifan Fan, Xibo Li, Ai Ti Aw, and Yu Hong. 2023e. GLGR: Question-aware global-to-local graph reasoning for multi-party dialogue reading comprehension. In *Findings of the EMNLP*, pages 1817–1826.

Shannan Liu, Peifeng Li, Yaxin Fan, and Qiaoming Zhu. 2025. Enhancing multi-party dialogue discourse parsing with explanation generation. In *Proceedings of the COLING*, pages 1531–1544.

Zhengyuan Liu and Nancy Chen. 2021. Improving multi-party dialogue discourse parsing via domain integration. In *Proceedings of the CODI*, pages 122–127.

Ryan Lowe, Nissan Pow, Iulian Vlad Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the SIGDIAL*, pages 285–294.

Tiezheng Mao, Jialing Fu, Osamu Yoshie, Yimin Fu, and Zhuyun Li. 2023. Matching intentions for discourse parsing in multi-party dialogues. In *Advances and Trends in Artificial Intelligence. Theory and Applications*, pages 130–140.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Proceedings of the NeurIPS*, pages 53728–53741.

Zhouxing Shi and Minlie Huang. 2019. A deep sequential model for discourse parsing on multi-party dialogues. In *Proceedings of the AAAI*, pages 7007–7014.

Kate Thompson, Akshay Chaturvedi, Julie Hunter, and Nicholas Asher. 2024. Llamipa: An incremental discourse parser. In *Findings of the EMNLP*, pages 6418–6430.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

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 IJCAI*, pages 3943–3949.

Ante Wang, Linfeng Song, Lifeng Jin, Junfeng Yao, Haitao Mi, Chen Lin, Jinsong Su, and Dong Yu. 2023a. D^2PSG: Multi-party dialogue discourse parsing as sequence generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31(1):4004–4013.

Chengrui Wang, Shaoming Ji, and Fang Kong. 2024. Local or global optimization for dialogue discourse parsing. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 149–161. Springer.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-Instruct: Aligning language models with self-generated instructions. In *Proceedings of the ACL*, pages 13484–13508.

Yunhe Xie, Chengjie Sun, Yifan Liu, Zhenzhou Ji, and Bingquan Liu. 2024. UniMPC: Towards a unified framework for multi-party conversations. In *Proceedings of the CIKM*, page 2639–2649.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Jingxuan Yang, Kerui Xu, Jun Xu, Si Li, Sheng Gao, Jun Guo, Nianwen Xue, and Jirong Wen. 2021. A joint model for dropped pronoun recovery and conversational discourse parsiin chinese conversational speech. In *Proceedings of the ACL-IJCNLP*, pages 1752–1763.

Nan Yu, Guohong Fu, and Min Zhang. 2022. Speaker-aware discourse parsing on multi-party dialogues. In *Proceedings of the COLING*, pages 5372–5382.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the ACL*, pages 400–410.

A Ambiguity Examples

Tables 4- 7 illustrate the examples of typos, abbreviations, slang, and idioms. In Table 4, the text highlighted in red is a typographical error, which should be corrected to "who wants sheep" based on the dialogue history. In Table 5, the red text is an abbreviation, where "u" stands for "you". In Table 6, the red text represents slang, with "cool" being a term that conveys agreement and confirmation. Lastly, in Table 7, the red text is an idiom, where "hats off to you" originally signifies a gesture of respect by removing one's hat. In this dialogue, it is used as an implicit expression of praise.

B Prompts

B.1 Input and Output Format for DP

The input format used to fine-tune the discourse parser for the example in Figure 2 is provided below:

> Below is a multi-party dialogue:
>
> u1, ztime: random 7 | u2, shawnus: damn | u3, ztime: doesn't happen like this in the real game does it...? | u4, somdechn: wood for clay? | u5, shawnus: two resources stolen! | u6, ztime: sorry...
>
> Please identify a dependency utterance for utterance u_6 and determine the rhetorical relationship between them.

Each utterance is indexed as u_i for a simplified representation of the output. The output format for the example is: "**u6, u5 : comment**", which indicates that the utterance $u6$ depends the utterance u_5 and their relation type is "comment.". If an utterance has no dependent utterance, the output is simply "none."

B.2 Clarification Data Construction Prompt

The prompt to generate the clarification data for the example in Figure 2 is shown below:

> Below is a multi-party conversation:
>
> u1, ztime: random 7 | u2, shawnus: damn | u3, ztime: doesn't happen like this in the real game does it...? | u4, somdechn: wood for clay? | u5, shawnus: two resources stolen! | u6, ztime: sorry...
>
> Let's break this down step by step.
>
> # Step 1: Evaluate whether u6 contains any **{"omission," "typo," "abbreviation," "slang," or "idiom."}**
>
> # Step 2: Follow the results of step 1 as a clarification direction and provide a clarified version of utterance u6 to ensure that **the comment relation type between utterance u6 and utterance u5 is clear** and **avoid the question-answer pair between utterance u6 and utterance u4.**
>
> Output Format:
> {
> "Step 1 Reasoning": "",
> "Step 2 Reasoning": "",
> "Clarified utterance": ""
> }
> Where:
> Step 1 Reasoning is the reasoning process for Step 1.
> Step 2 Reasoning is the reasoning process for Step 2.
> Clarified utterance is the clarified version of utterance u6.

B.3 Input and Output Format for DCM

The input format to fine-tune DCM for the example in Figure 2 is shown below:

> Please clarify the last utterance:
>
> u1, ztime: random 7 | u2, shawnus: damn | u3, ztime: doesn't happen like this in the real game does it...? | u4, somdechn: wood for clay? | u5, shawnus: two resources stolen! | u6, ztime: sorry...

And the output format is "CTR, DGR, u_{ck}", where CTR, DGR and u_c denote the clarification type reasoning, discourse goal reasoning, and the clarified utterance, respectively.

C Manual Analysis of Clarification Types

We conducted a manual analysis to identify the primary types of clarifications required to improve

Typo	
Dialogue History	u1, somdechn: 12 aagain... u2, ztime: dude.. u3, shawnus: haha you are far ahead! u4, somdechn: who whats sheep?
Intended Discourse Relation	u4, u1 : continuation
Ambiguous Discourse Relation	u4, u2 : clarification_question
Clarification of u4	Who wants sheep?

Table 4: An example of a typo.

Abbreviation	
Dialogue History	u1, william: hi markus. ... u14, william: the arrow is pointing at me u15, william: but i cant press roll u16, william: oh sorry u17, thomas: u can place a settlement u18, thomas: first u19, thomas: u roll later
Intended Discourse Relation	u19, u18 : narration
Ambiguous Discourse Relation	u19, u18 : continuation
Clarification of u19	you roll later.

Table 5: An example of an abbreviation.

discourse parser predictions. For this study, a random sample of 500 instances where the discourse parser made incorrect predictions on the validation set was selected. The analysis was conducted by a team of three NLP researchers, including one PhD candidate and two graduate students, all of whom possess expertise in dialogue discourse parsing. They independently examined the linguistic features present in the utterances that could potentially lead to ambiguous understanding by the discourse parser and voted on the final clarification types. As a result, five primary types of clarification were identified: omission, typo, abbreviation, slang, and idiom. Detailed statistics for the STAC and Molweni datasets are illustrated in Figure 6. Omissions constitute the largest proportion, a common linguistic feature in conversations. Additionally, even when the utterance is formally expressed, the discourse parser can still make errors, with a proportion of 7% in STAC and 11% in Molweni. In this paper, we focus on addressing these five informal linguistic features to significantly enhance the performance of discourse parsers.

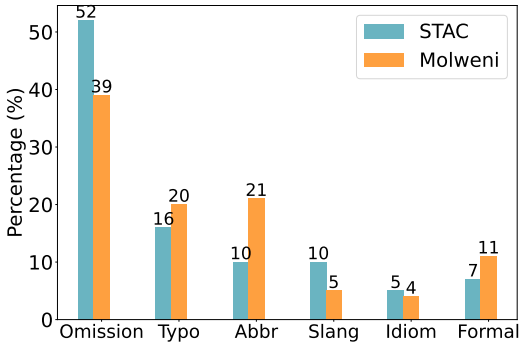

Figure 6: Results of the manual analysis on STAC and Molweni. "Abbr" denotes "Abbreviation".

D Analysis of Clarification Data Volume

Our method allocated α % of the training set to construct the clarification data for fine-tuning DCM, while the remaining 1-α % was used for preference optimization to enhance DCM's adaptation to DP. Figure 7 and 8 illustrate the impact of varying clarification data volumes on STAC and Molweni. Notably, increasing the volume of data for fine-tuning (see DP-DCM) did not significantly enhance the parser's performance. This may be attributed to the additional noise introduced by larger

Slang	
Dialogue History	u1, gaeilgeoir: well played u2, inca: cheers, good game u3, nareik15: nice. good game u4, gaeilgeoir: talk soon u5, inca: shall we say wednesday for the one without kieran? u6, gaeilgeoir: sounds fine to me u7, gaeilgeoir: time? u8, inca: cool, any time's fine for me, 8 again? u9, gaeilgeoir: yay u10, gaeilgeoir: can't wait u11, inca: cool, see you then!
Intended Discourse Relation	u11, u9 : acknowledgement
Ambiguous Discourse Relation	u11, u8 : result
Clarification of u11	Looking forward to it, see you then!

Table 6: An example of slang.

Idiom	
Dialogue History	u1, somdechn: :) u2, ztime: :-) u3, ztime: thanks!!!! u4, shawnus: damn! u5, somdechn: nice one bro... u6, shawnus: nice one u7, ztime: that was a close game.... u8, shawnus: yeah u9, shawnus: hats off to you
Intended Discourse Relation	u9, u7 : comment
Ambiguous Discourse Relation	u9, u8 : continuation
Clarification of u9	you played really well

Table 7: An example of an idiom.

volumes of clarification data. Conversely, incorporating CPO at various data volumes improved the parser's performance. However, as the proportion of preference data decreased, the effectiveness of CPO diminished. Our method achieved optimal performance with α set 10% and 20% on STAC and Molweni, respectively.

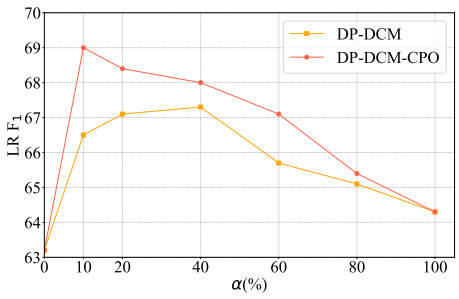

Figure 7: The LR F_1 performance of using varying volumes of clarification data on STAC.

E Implementation Details

The training hyper-parameters for DP and DCM were kept consistent, as detailed in Table 8. The hyper-parameters for preference optimization of DCM are listed in Table 9.

F Experiments with Different Backbones and Parameter Sizes

To demonstrate the versatility of our approach, we employed Qwen2 (Yang et al., 2024) as our

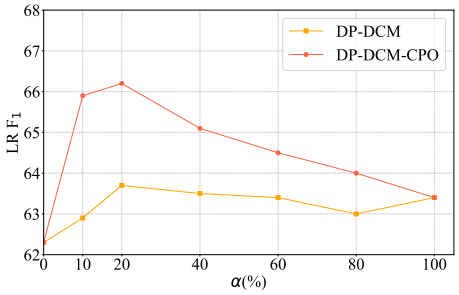

Figure 8: The LR F_1 performance of using varying volumes of clarification data on Molweni.

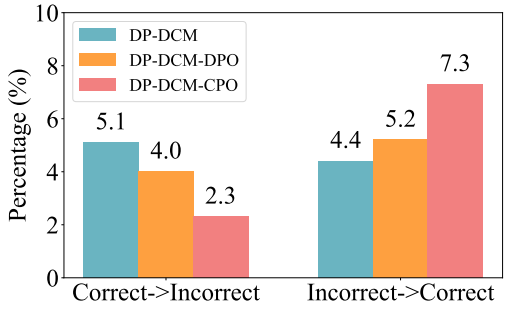

Figure 9: Comparison of our CPO with standard DPO on Molweni in two scenarios.

Parameter	Value
learning rate	1e-4
batch size	1
gradient accumulation steps	8
epoch	3
warmup ratio	0.1
bf16	True
optimizer	AdamW
sequence length	1024

Table 8: Hyperparameter settings in the fine-tuning stage.

Parameter	Value
learning rate	5e-6
batch size	1
gradient accumulation steps	8
epoch	1
warmup ratio	0.1
bf16	True
optimizer	AdamW
sequence length	1024
β	0.1
μ	0.7/0.5

Table 9: Hyperparameter settings during the preference optimization stage. The μ values for STAC and Molweni are set to 0.7 and 0.5, respectively, determined by a grid search within {0.1, 1.0}.

backbone model, utilizing both the 1.5B[8] and 7B[9] versions, and performed parameter-efficient fine-tuning using LoRA. The experimental results are presented in Table 10. Remarkably, our method significantly outperforms the previous SOTA model, DDPE, even with the smaller 7B parameter configuration, highlighting the efficiency of our approach. Furthermore, although model performance tends to decline with a smaller parameter size, our 1.5B model achieves performance comparable to DDPE with 8B parameters. Additionally, the removal of DCM (along with CPO, which is used to optimize DCM) leads to a significant drop in model performance. These results strongly validate the versatility and effectiveness of our method.

[8] https://huggingface.co/Qwen/Qwen2-1.5B-Instruct

[9] https://huggingface.co/Qwen/Qwen2-7B-Instruct

G Analysis

G.1 Analysis of DCM on Molweni

Table 11 illustrates the performance degradation on Molweni across different clarification types, which is caused by the removal of CTR and DGR. Consistent with the trend observed in STAC, DCM primarily addresses omission, the largest source of errors. Removing DTR more strongly affects abbreviation, slang, and idiom, while removing DGR significantly impacts omission. Together, DTR and DGR complement each other, improving the overall robustness and clarification capability of DCM.

G.2 Analysis of CPO on Molweni

The distribution on Molweni is illustrated in Figure 9. We observed that 5.1% of DP's initially correct predictions became incorrect after DCM clarification when CPO was removed. By improving the adaptability of DCM to DP using DPO, this Correct->Incorrect proportion was effectively reduced to 4.0%. Notably, CPO enhances DCM by capturing the contribution gaps of preference pairs, reducing the Correct->Incorrect proportion to 2.3%. Furthermore, CPO also increases the Incorrect->Correct proportion from 4.4% to 7.3%, compared to DPO's 5.2%. These observations align with the patterns seen in STAC, further demonstrating the effectiveness of our CPO method.

	Model	LM	STAC		Molweni	
			L F_1	LR F_1	L F_1	LR F_1
Generative	DDPE † (SOTA)	LLaMA3 (8B)	79.5	63.4	87.6	62.9
	DP-DCM-CPO†	Qwen2 (7B)	80.6	66.1	87.9	64.7
	w/o DCM&CPO		76.5	61.9	86.2	61.4
	DP-DCM-CPO†	Qwen2 (1.5B)	78.9	63.8	86.9	63.1
	w/o DCM&CPO		75.9	58.8	85.2	60.6

Table 10: Experimental results with different parameter sizes for Qwen2 backbone on STAC and Molweni., where † represents parameter-efficient fine-tuning with LoRA.

Category	Omission	Typo	Others
Percentage(%)	50	27	23
Accuracy(%)			
DCM	24.0	13.7	8.6
w/o CTR	23.4	13.2	3.2
w/o DGR	20.1	13.0	6.3

Table 11: Performance degradation on Molweni across different types caused by the removal of CTR and DGR. The category "Others" includes abbreviation, slang, and idiom.

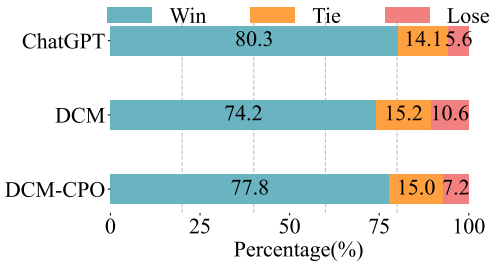

Figure 10: Results of the human pairwise evaluation of clarification data quality on Molweni.

G.3 Quality Analysis of Clarification Data

We randomly selected 500 validation samples for the manual evaluation of ChatGPT, DCM, and DCM-CPO. One PhD candidate and two graduate students, experience in annotating discourse relations under the Segmented Discourse Representation Theory (SDRT) (Asher and Lascarides, 2003), independently assessed whether the clarified or original utterances more clearly conveyed the intended relation types with their dependent utterances. The final results were determined by majority vote. Both the clarified and original utterances were randomly shuffled and anonymized to ensure unbiased evaluation.

The results on Molweni are illustrated in Figure 10. It was observed that 80.3% of the utterances clarified by ChatGPT were superior, while 5.6% were inferior to the original utterances. Our DCM, which is trained on data from ChatGPT, generated 74.2% superior and 10.6% inferior clarifications compared to the original utterances. Furthermore, our DCM-CPO, which optimizes DCM by leveraging parser feedback to mitigate noise, resulted in 77.8% superior and 7.2% inferior clarifications compared to the original utterances. These observed patterns were consistent with the results obtained on the STAC dataset.

G.4 Case Study

Table 4 illustrates an ambiguity caused by a typographical error. In u_4, the term "whats" is a typo, which led DP to incorrectly identify a clarification_question relation type between u_4 and u_2. Our DCM correctly identified and corrected "whats" to "wants", enabling DP to parse correctly.

Table 5 demonstrates an ambiguity resulting from an abbreviation. In u_{19}, "u" is an abbreviation for "you", which caused the parser to erroneously identify a narration relation type between u_{19} and u_{18}. To resolve this, our DCM clarified "u" to "you", allowing the parser to perform accurate parsing.

Table 6 presents an ambiguity caused by slang. In u_{11}, "cool" is a slang term implicitly expressing agreement or confirmation, leading the parser to incorrectly identify a result relation type between u_{11} and u_8. To address this, Our DCM understood the implicit meaning of "cool" and clarified it to "looking forward to it," enabling the parser to parse accurately.

Table 7 highlights an ambiguity caused by an idiom. In u_9, "hats off to you" is an idiom which is used to express praise. To resolve this, our DCM comprehended its implicit meaning and clarified it to "you played really well," allowing the parser to perform accurate parsing.

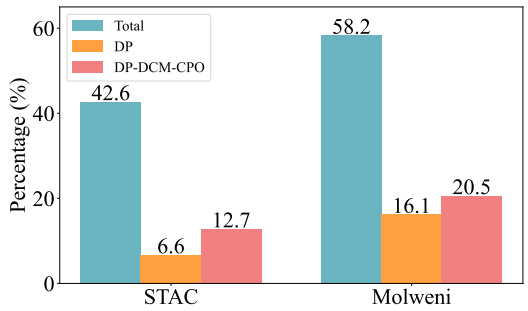

Figure 11: Percentage of uncertain instances on STAC and Molweni.

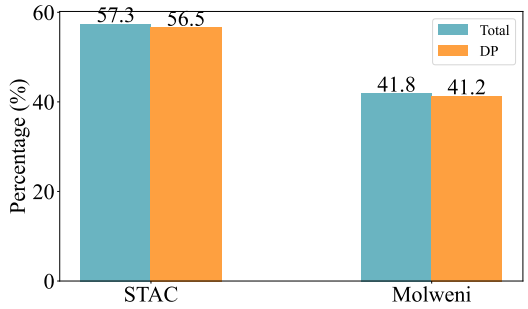

Figure 12: Percentage of certain instances on STAC and Molweni.

G.5 Analysis of Uncertainty Assessment

The uncertainty assessment process focuses on identifying instances where DP exhibits uncertainty during prediction. These instances are processed by DCM for clarification. Details are provided in Section 4.4. To evaluate the effectiveness of this approach, we analyzed the percentage of uncertain instances and the accuracy of predictions made by DP and DP-DCM-CPO. The results are shown in Figure 11.

On the STAC dataset, 42.6% of instances were identified as uncertain, with only 6.6% correctly predicted by DP. Similarly, on the Molweni dataset, 58.2% of instances were classified as uncertain, with 16.1% correctly predicted by DP. These results show that DP's accuracy drops significantly when it lacks confidence in its predictions. However, our DP-DCM-CPO provides clarifications for these uncertain instances, improving prediction accuracy. Specifically, the correct prediction rate for uncertain instances increased from 6.6% to 12.7% on STAC and from 16.1% to 20.5% on Molweni.

The percentage of certain instances is shown in Figure 12. On STAC, 57.3% of instances were identified as certain, with 56.5% correctly predicted by DP. On Molweni, 41.8% of instances were classified as certain, with 41.2% correctly predicted by DP. This indicates that DP rarely makes errors when confident in its predictions, making further clarification unnecessary for these instances.

In summary, these findings demonstrate that our uncertainty assessment method effectively distinguishes between uncertain and certain instances, enabling targeted improvements in prediction accuracy and overall parsing performance.