Coherence-based Dialogue Discourse Structure Extraction using Open-Source Large Language Models

Gaetano Cimino[1*], Chuyuan Li[2], Giuseppe Carenini[2], Vincenzo Deufemia[1]
[1]University of Salerno, 84084, Fisciano, Salerno, Italy
[2]University of British Columbia, V6T 1Z4, Vancouver, BC, Canada
{gcimino, deufemia}@unisa.it
chuyuan.li@ubc.ca, carenini@cs.ubc.ca

Abstract

Despite the challenges posed by data sparsity in discourse parsing for dialogues, unsupervised methods have been underexplored. Leveraging recent advances in Large Language Models (LLMs), in this paper we investigate an unsupervised coherence-based method to build discourse structures for multi-party dialogues using open-source LLMs fine-tuned on conversational data. Specifically, we propose two algorithms that extract dialogue structures by identifying their most coherent sub-dialogues: DS-DP employs a dynamic programming strategy, while DS-FLOW applies a greedy approach. Evaluation on the STAC corpus demonstrates a micro-F_1 score of 58.1%, surpassing prior unsupervised methods. Furthermore, on a cleaned subset of the Molweni corpus, the proposed method achieves a micro-F_1 score of 74.7%, highlighting its effectiveness across different corpora.

1 Introduction

Understanding multi-party dialogue structure is crucial for various natural language tasks like dialogue comprehension, summarization, and sentiment analysis (Joty et al., 2019; Li et al., 2019; He et al., 2021; Feng et al., 2022). The goal is to extract a coherent discourse structure from a dialogue transcript, wherein pairs of clause-like texts are linked through rhetorical relations. To obtain a good understanding of the coherent discourse structures in dialogues, the Segmented Discourse Representation Theory (SDRT) (Asher and Lascarides, 2003) framework proposes to annotate dialogues with dependency graphs, where edges link text spans labeled with semantic-pragmatic relations. An example dialogue derived from the Strategic Conversations corpus (STAC) (Asher et al., 2016) corpus is shown in Figure 1, with

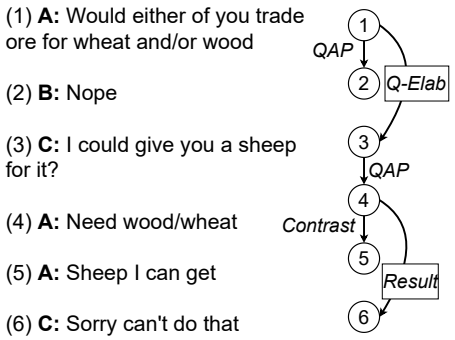

Figure 1: A dialogue instance from the STAC corpus (id *pilot04_6*), illustrating user utterances on the left and the corresponding ground-truth dialogue structure and relations on the right. The graph reveals three distinct sub-dialogues: $(1, 2)$, $(1, 3, 4, 5)$, and $(1, 3, 4, 6)$.

nodes denoting discourse units and edges relation types (i.e., *Question-Answer Pair* (QAP), *Question-Elaboration* (Q-Elab), *Contrast*, and *Result*).

Multi-party dialogues pose greater challenges compared to two-party dialogues, due to the involvement of numerous speakers, each contributing uniquely with more speech turn interactions and structural particularities (Asher et al., 2016). Nevertheless, this complexity allows for the segmentation of dialogues into independent conversational flows that share a common overarching topic. These conversational flows exhibit distinct internal progression and structure, thereby permitting them to be regarded as sub-dialogues (Fernández et al., 2008; Frampton et al., 2009; Sun et al., 2016). As a result, the discourse structure of multi-party dialogues can be predicted by decomposing dialogues into coherent sub-dialogues, where each sub-dialogue reflects the flow of conversation, starting with an initial utterance and concluding when no further elaboration occurs. For instance, the dialogue in Figure 1 comprises three sub-dialogues: (1, 2), (1, 3, 4, 5), and (1, 3, 4, 6). However, exploring all possible sub-dialogues to identify the coherent ones is unrealistic because it involves ana-

[*] This work was done during a visit to the University of British Columbia.

Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 297–316
September 18–20, 2024.

lyzing all possible ordered sequences of utterances within a dialogue, leading to exponential growth that makes exhaustive analysis impractical.

Supervised evaluation of dialogues is challenging due to data sparsity (Li et al., 2022). To address this issue, some studies have proposed unsupervised (Li et al., 2023) and semi-supervised (Badene et al., 2019b,a; Nishida and Matsumoto, 2022; Li et al., 2024a) methods. These methods typically predict the best discourse pair given a discourse unit, while overlooking the previous context. However, we advocate that identifying sub-dialogues can offer a broader context to better understand the thematic coherence within a dialogue, thus building a more accurate discourse structure.

In this paper, we propose an unsupervised, sub-dialogue-oriented method for extracting "naked" discourse structures without discourse relations in multi-party dialogues. Although without relations, discourse structures alone have been shown to be crucial features for tasks such as content selection (Louis et al., 2010) and summarization (Xiao et al., 2020; Xu et al., 2020). Precisely, we introduce two algorithms: Multi-Party Dialogue Structure Extraction based on Dynamic Programming (DS-DP) and Multi-Party Dialogue Structure Extraction based on Flow Conversation Analysis (DS-FLOW), designed to decompose dialogues into coherent sub-dialogues. DS-DP identifies the most coherent (partial) sub-dialogues ending in each discourse unit by applying a dynamic programming strategy. In contrast, DS-FLOW greedily predicts for each discourse unit the most likely coherent subsequent utterances, followed by a process that ensures the completeness of the resulting discourse structure. In both algorithms, we use perplexity as a metric to evaluate sub-dialogue coherence. To compute perplexity scores, we draw inspiration from work on Pre-trained Language Models (PLMs) and Large Language Models (LLMs) fine-tuned on conversational data implicitly capturing dialogue quality (Mehri and Eskénazi, 2020; Bruyn et al., 2022).

We utilize open-source models, as proprietary models are limited to text-based prompts and do not permit analysis of output probabilities. In practice, we compare the performance of two open-source LLMs: a chatbot trained by fine-tuning LLaMA on user-shared conversations Vicuna-13B (Chiang et al., 2023) and a general-purpose model Mistral-7B (Jiang et al., 2023). We evaluate our method on the STAC corpus (Asher et al., 2016) and a revised subset of the Molweni corpus (Li et al., 2020). The results demonstrate the effectiveness of our solution, as it outperforms prior unsupervised methods. Specifically, we achieve a micro-F_1 score of 58.1% on STAC and 74.7% on Molweni, demonstrating its robustness across different corpora.

The contributions of this paper are twofold. First, we propose a fully unsupervised method for extracting graph structures of multi-party dialogues, which is the first of its kind to the best of our knowledge. Second, we introduce and evaluate two novel algorithms that leverage open-source LLMs to decompose dialogues into coherent sub-dialogues, enabling a more fine-grained analysis of discourse structures.

2 Related Work

Multi-Party Dialogue Discourse Parsing Various methodologies have been proposed for parsing multi-party dialogues. Perret et al. (2016) developed an Integer Linear Programming approach predicting non-tree structures by encoding linguistic principles as constraints. Wang et al. (2021) presented the Structure Self-Aware model, using an edge-centric graph neural network to learn representations of discourse unit pairs directly. Bennis et al. (2023) introduced BERTLine, a discourse parsing model leveraging a multi-task setup to jointly predict discourse attachments and relation labels, achieving state-of-the-art performance. Mao et al. (2024) proposed the Hierarchical Graph Fusion Network, using hierarchical graph neural networks to encode contextual levels like utterances, dialogue topics, and user preferences. While effective, these approaches rely on annotated data, posing challenges due to limited resources. To address data sparsity, recent studies have explored unsupervised and semi-supervised strategies using PLMs and LLMs. For instance, Li et al. (2023) proposed extracting dependency trees from PLM attention matrices using unsupervised metrics or semi-supervised strategies with small validation sets. Instead, Li et al. (2024a) designed a semi-supervised pipeline to predict structures and relations sequentially via self-training. In another study, Chan et al. (2023) used zero- and few-shot prompting techniques to assess ChatGPT on discourse parsing, but achieved abysmal results. In contrast, the method proposed in this paper requires no annotation, using fully unsupervised approaches to extract discourse structures in multi-party dialogues. Furthermore, unlike the unsupervised ap-

proach proposed by Li et al. (2023), our solution can also extract graph structures rather than being limited to dependency trees.

LLMs for Dialogue Evaluation Prior research has highlighted the inherent ability of PLMs and LLMs to implicitly capture dialogue quality, making them suitable for evaluating dialogues. Mehri and Eskénazi (2020) and Bruyn et al. (2022) introduced the FED and FULL metrics, respectively, to assess open-domain dialogue systems utilizing PLMs and LLMs without requiring ground-truth responses or supervised training data. These metrics evaluate dialogue quality by estimating the likelihood of a model generating follow-up utterances aligned with different dimensions of dialogue quality after a given system response. The strong correlation observed between metric scores and human judgments suggests that PLMs and LLMs have acquired meaningful representations of dialogue quality aligned with human perceptions. Similarly, Zhang et al. (2024) analyzed LLMs as automatic dialogue evaluators, inspired by the remarkable performance of LLMs fine-tuned using the instruction-tuning approach (Zhang et al., 2023). Their study involved multidimensional evaluation of proprietary and open-source LLMs for assessing dialogue quality across various dimensions. Results indicate that appropriately aligned and utilized LLMs can effectively serve as generalized automatic dialogue evaluators, complementing human judgments. Motivated by these findings, in this paper we evaluate sub-dialogues by leveraging LLMs' capabilities in generating coherent dialogues and adhering to relevant instructions.

Sub-Dialogue Detection Sub-dialogues are extensively studied in computational tasks, notably within Dialogue State Tracking (DST) (Sun et al., 2016; Lee et al., 2021), aiming to understand decisions in multi-party conversations. In this framework, a dialogue session is deconstructed into a series of sub-dialogues, each consisting of consecutive multi-turn exchanges focused on a shared topic. Departing from conventional DST approaches, in this paper we adopt sub-dialogue delineation to extract the underlying structure of multi-party dialogues. Specifically, our method involves unsupervised evaluation of multi-party dialogue discourse units, leveraging insights from LLMs fine-tuned on conversational data.

3 Method

In this section, we first formally describe the dialogue parsing task. We then describe two algorithms for sub-dialogue extraction. The first relies on a dynamic programming strategy and is formally denoted as Multi-Party Dialogue Structure Extraction based on Dynamic Programming (DS-DP). Conversely, the second algorithm, grounded in the analysis of conversation flows, is formally named Multi-Party Dialogue Structure Extraction based on Flow Conversation Analysis (DS-FLOW).

3.1 Problem Formulation

Let $D = (e_1, e_2, \ldots, e_n)$ be a dialogue consisting of *n Elementary Discourse Units* (EDUs), each representing the smallest unit of discourse. In the SDRT framework, a dialogue D can be represented as a *Directed Acyclic Graph* (DAG), denoted as $DAG(D)$, wherein EDUs are connected with directed edges. Dialogue discourse parsing aims to automatically derive a DAG that best represent the SDRT structure of a dialogue. In our proposal, the construction of $DAG(D)$ involves creating m sub-dialogues $Sub_1(D), \ldots, Sub_m(D)$, where each sub-dialogue possesses an intrinsic structure $S_{Sub_j(D)} \subseteq DAG(D)$. Thus, discourse structure extraction can be reframed as linking EDUs within D to form the most coherent sub-dialogues, such that $DAG(D) = \bigcup_{j=1}^{m} S_{Sub_j(D)}$. Note that the following properties hold:

- We assume the absence of backward links in the final DAG as an utterance cannot depend, either anaphorically or rhetorically, on subsequent utterances within a dialogue, as they are previously unknown (Afantenos et al., 2012; Li et al., 2023).

- Each sub-dialogue must include an edge originating from the initial EDU e_1. This means that:

  $$\forall j \in \{1, \ldots, m\}\ \exists (e_1, e_k) \in S_{Sub_j(D)}$$

  This constraint is justified by the fact that all EDUs, except e_1, must have at least one incoming edge from a previous node, and recursively following these edges backward leads to e_1. In SDRT, the DAGs have unique roots, so that every single EDU is reachable from the first EDU, i.e., the axiom (Perret et al., 2016).

- Sub-dialogues can overlap, allowing certain edges to be part of multiple sub-dialogues, justified by the fact that speaker interventions may

contribute to different themes within one dialogue. For instance, the edge (e_3, e_4) in Figure 1 appears in two sub-dialogues. In sub-dialogue (e_1, e_3, e_4, e_5), this edge leads to speaker A's elaboration in e_5 on their inquiry in e_4, which was prompted by speaker C's question in e_3. In sub-dialogue (e_1, e_3, e_4, e_6), the edge (e_3, e_4) leads to speaker C's declination in e_6 of speaker A's inquiry in e_4.

3.2 DS-DP Algorithm

This algorithm uses dynamic programming to efficiently explore the space of all possible sub-dialogues within a dialogue. As a first step, given a dialogue D as input, the algorithm maps it into a fully-connected graph with only forward links $G = (V, E)$. In this graph, V represents the set of EDUs within the dialogue, and E includes all potential links in the dialogue's structure. The DS-DP algorithm aims to extract from G the paths corresponding to the most coherent (partial) sub-dialogues starting from the initial EDU and ending in each subsequent EDU, based on a coherence metric denoted as *eval*. To this end, it defines two matrices of size $(|\mathrm{V}|-1)\times(|\mathrm{V}|-1)\times(|\mathrm{V}|-2)$, which we call M_{co} and M_{pred}. Here, $M_{\text{co}}[i][j][k]$ denotes the maximum coherence of a sub-dialogue passing through node i, ending in node j, with k preceding nodes before node i. Similarly, $M_{\text{pred}}[i][j][k]$ stores the previous node to achieve the maximum coherence value of the sub-dialogue ending in node j, passing through node i, and considering k preceding nodes before node i. Taking the unidirection property into account, only the upper right half of the matrices contain valid values; no values are stored in the lower left part of the matrices. For initialization, the assignment

$$\forall j \quad M_{\text{co}}[0][j][0] = eval(e_1, e_j)$$

is set, rooted in the recognition that the only sub-dialogues without preceding nodes are those progressing from the initial node to any subsequent node. As a result, for each EDU e_i $(i > 1)$, the algorithm computes the most coherent sub-dialogues starting from e_1 and ending in e_i, with k intermediate nodes $(k \in [1, i-1])$. Specifically, each EDU e_i may either directly connect to e_1 or include up to $i-1$ edges within its most coherent sub-dialogue.

The pseudo-code of the DS-DP algorithm for matrix construction is presented in Algorithm 1. It iterates through each k value within the range from 1 to $|\mathrm{V}|-2$. For each k, it systematically traverses each node i according to the topological order defined on G. Subsequently, for each node i, it explores all possible successor nodes j. During this traversal, it examines each node u that has an edge directed towards i. The condition $M_{\text{co}}[u][i][k-1] \neq NULL$ indicates that node u has been previously visited, implying the feasibility of reaching node i from u by considering $k-1$ preceding nodes along the path. This condition ensures the consideration of only those nodes u that are accessible from i and can therefore serve as intermediary nodes to reach j with $k-1$ previous nodes. Upon satisfying this condition, the algorithm evaluates the coherence of the sub-dialogue ending at j, including $k-1$ preceding nodes, u, and i. If this assessment yields a coherence value superior to the one currently stored in $M_{\text{co}}[i][j][k]$, the matrix is updated with the new coherence value, and the predecessor information is recorded in $M_{\text{pred}}[i][j][k]$. To identify a sub-dialogue starting from the initial node and ending in a specified node e_j, it is sufficient to examine all non-null entries in the M_{co} matrix while keeping j constant. Subsequently, the sub-dialogue characterized by the highest coherence value is considered. The final DAG is then constructed by combining the identified sub-dialogues. An illustration of the application of DS-DP to the dialogue depicted in Figure 1 is provided in Appendix E.

Algorithm 1 DS-DP - Matrix Construction

Input: $G = (V, E)$
Output: Updated M_{co} matrix and M_{pred} matrix

for $k \leftarrow 1$ **to** $|V| - 2$ **do**
  for i **in** V **do**
    for j **in** V **do**
      if $j > i$ **then**
        for each node u **with an edge into** i **do**
          if $M_{\text{co}}[u][i][k-1] \neq$ NULL **then**
            $val \leftarrow eval(k-1 \text{ EDUs}, u, i, j)$
            if val better than $M_{\text{co}}[i][j][k]$ **then**
              $M_{\text{co}}[i][j][k] \leftarrow val$
              $M_{\text{pred}}[i][j][k] \leftarrow u$
end of all loops and conditions

From a complexity analysis perspective, the DS-DP algorithm comprises four nested loops for matrix construction and two nested loops for structure prediction. The first three outermost loops in Algorithm 1 iterate *O(|V|)* times each, resulting in a time complexity of *O(|V|^3)*. The last innermost loop processes all incoming edges of the current node, which has a time complexity of *O(|V|)*. For structure prediction, each node j requires iteration over $j-1$ values (since each path ending in node j can have a maximum length of $j-1$), resulting in

a time complexity of $O(|V|^2)$. Consequently, the overall time complexity of DS-DP is dominated by the matrix construction process, which has a worst-case time complexity of $O(|V|^4)$.

Coherence Evaluation Coherence, defined by the seamless flow and logical progression inherent in conversational interactions, stands as a pivotal criterion for assessing dialogues. Within the text analysis context, *perplexity* emerges as a valuable metric for evaluating the coherence of textual constructs (Colla et al., 2022). Consequently, we adopt perplexity as the *eval* metric to quantitatively measure how effectively a sub-dialogue maintains its logical structure and natural progression. Drawing from earlier studies indicating that LLMs capture elements of dialogue quality (Mehri and Eskénazi, 2020; Bruyn et al., 2022), we employ them to estimate the joint probability of each sub-dialogue $Sub_D = (e_1, \ldots, e_l)$ of a dialogue D. The perplexity score is calculated as

$$Pe(Sub_D) = \exp\left(-\frac{1}{l}\sum_{i=1}^{l}\log P(e_i|e_{<i})\right)$$

and provides insights into the model's level of certainty or uncertainty in predicting the unfolding discourse. Lower perplexity scores indicate higher coherence, demonstrating the model's proficiency in comprehending the logical flow of conversation.

3.3 DS-FLOW Algorithm

While DS-DP constructs sub-dialogues by identifying the most likely antecedents of a given EDU, DS-FLOW mainly focuses on capturing the most fluent successive utterances of a given EDU. Specifically, it consists of three steps: (i) In the first step, for each EDU excluding the final one, the algorithm predicts the most probable subsequent EDU that elaborates upon it. Notably, previous incoming links to EDUs are utilized to inform these predictions. We evaluate sub-dialogue coherence using the perplexity metric, as discussed in DS-DP. (ii) In the second step, a filtering mechanism is applied to recognize the conclusion of conversational segments. This step addresses the issue that not all utterances are elaborated upon further, resulting in certain nodes lacking outgoing links. (iii) The third step involves a backward analysis to address potential *orphan* EDUs (i.e., EDUs without incoming edges) due to the filtering process or the lack of links predicted in the first step. For each orphan EDU e_i, the analysis selects a parent out of all sub-dialogues ending in an EDU e_j where $i > j$.

An illustration of the application of DS-FLOW to the dialogue depicted in Figure 1 is provided in Figure 2. It elucidates the following steps: the initial identification of an outgoing link for each EDU, the subsequent filtration of links (e_2, e_4) and (e_5, e_6), and the selection of sub-dialogues (e_1, e_3) and (e_1, e_3, e_4, e_5) in the backward analysis due to the absence of incoming links for e_3 and e_5 in the first two steps, thereby augmenting the final DAG with edges (e_1, e_3) and (e_4, e_5).

From a complexity analysis perspective, the DS-FLOW algorithm constructs a DAG with $|V| = n$ nodes from a dialogue containing n EDUs through three sequential steps. Initially, it predicts the subsequent EDU for each dialogue segment by leveraging prior connections, achieving a linear time complexity of $O(|V|)$. Following this, it filters segments that terminate without additional elaboration, also operating in linear time $O(|V|)$. Subsequently, in its third step, DS-FLOW undertakes a backward analysis to assign appropriate parents to orphan EDUs from previously identified sub-dialogues. In the worst-case scenario, this involves evaluating each orphan against all preceding EDUs, resulting in a time complexity of $O(|V|^2)$. Consequently, the overall time complexity of DS-FLOW is $O(|V|^2)$.

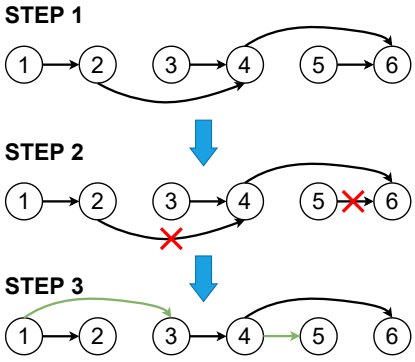

Figure 2: An example of DS-FLOW execution.

Filtering Mechanism An approach to implementing the filtering mechanism entails employing instruction-tuned LLMs as automatic dialogue evaluators (Zhang et al., 2024), prompting these models to generate responses for potential dialogue continuations. Specifically, given a pair of EDUs, the LLM is tasked with evaluating whether the second EDU (i.e., the next sequential EDU in the dialogue) builds upon the first one. However, as noted by Zhang et al. (2024) and confirmed through our own experimentation, the text generated by open-

source LLMs can become problematic, featuring content that is nonsensical or inaccurate (Rawte et al., 2023). Consequently, we follow the method outlined by Gupta et al. (2022), employing an implicit scoring mechanism. Specifically, when presented with an instruction prompt input[1], we focus on the output probabilities associated with the words "Yes" and "No" as generated by the LLM. In this context, we compute the probability

$$P(e_i \rightarrow e_j) = \frac{P(Finaltoken=\text{“}Yes\text{”})}{P(Finaltoken=\text{“}Yes\text{”})+P(Finaltoken=\text{“}No\text{”})}$$

where we evaluate the likelihood of having an EDU e_j as a subsequent utterance following the EDU e_i. In the DS-FLOW algorithm, we discard outgoing links with a probability lower than 0.5. The evaluation of the filtering mechanism's performance revealed that it filters a limited number of links with commendable reliability. Additional details are provided in Appendix C.

3.4 Additional Constraints

The proposed approaches for analyzing sub-dialogues within a dialogue face a challenge of preserving semantic coherence. Specifically, certain sub-dialogues may lack coherence, such as examining the link between the first and last EDUs in a long dialogue, which is unlikely to constitute a valid connection. To illustrate this challenge, consider the following dialogue excerpt:

(e_1) **A**: Did you enjoy the movie last night?
(e_2) **B**: Yeah, the plot twist was unexpected.
. . .
(e_{p-1}) **A**: What did you think about the ending?
(e_p) **B**: Oh, it was great!

where p is a large number. In this scenario, a valid link exists between e_1 and e_2. However, when examining individual pairs of EDUs, e_p may erroneously be deemed as coherent with e_2 in relation to e_1, despite their temporal separation and semantic incongruity within the ongoing conversation. To mitigate the issue of incoherent sub-dialogues, we advocate for including a hard constraint on the distance between two EDUs under scrutiny. As done by Bennis et al. (2023), when assessing the potential linkage between an EDU e_j and one of its preceding e_i, we impose the condition $j > i \geq j - 10$. By analyzing the development sets from the STAC and Molweni corpora, we observed that fewer than 1.9% of the links fail to meet the specified condition. By limiting the distance between EDUs, we reduce computational complexity and enhance the likelihood of extracting relevant information from nearby EDUs, thereby improving the coherence of sub-dialogues. Additionally, we propose integrating a penalization factor $P_{dist}(d)$, where d represents the number of intervening speech turns between two EDUs. This factor increases the perplexity associated with a sub-dialogue as the temporal distance between the two EDUs to be linked increases. By prioritizing proximity between EDUs, the incorporation of the penalization factor aims to account for the potential degradation of coherence over time. Specifically, we adopt $\sqrt{d}$ as the penalization factor for perplexity scores. This penalty is applied by multiplying the perplexity score of a sub-dialogue by the output of $P_{dist}(d)$.

[1]The prompt is detailed in Appendix B.

4 Experimental Setup

Corpora We conduct experiments on two commonly used SDRT-annotated dialogue corpora: (i) **STAC** (Asher et al., 2016). This corpus contains 1161 multi-party dialogues arising from interactions within an online version of the game "The Settlers of Catan". Given the unsupervised nature of our method, we evaluate it on the test set, which consists of 109 documents, amounting to 1129 EDU pairs. (ii) **Molweni** (Li et al., 2020). Derived from the Ubuntu Chat Corpus (Lowe et al., 2015), this corpus centers around technical discussions concerning the Ubuntu system. Due to quality issues with the original annotations (Li et al., 2023), we employ the "**Molweni-clean**" version proposed by Li et al. (2024a), which consists of 50 documents, encompassing 373 EDU pairs. Detailed corpus statistics are presented in Table 1.

Evaluation Metrics To assess the performance of the proposed approaches, we report the micro-F_1, recall, and precision for the generated structures.

Compared Methods We contrast our method with the straightforward yet strong unsupervised LAST baseline (Schegloff, 2007), which links each EDU with the preceding one. Moreover, we compare it with the method proposed by Li et al. (2023), currently the only known unsupervised approach in the literature proficient at predicting discourse structure, albeit without explicitly extracting DAGs. Finally, to draw insights from modern LLMs, we present results from ChatGPT (*gpt-3.5-turbo ver-*

Corpus	#Doc	#Turn/doc	#Tok/doc	#Spk/doc
STAC	109	10.6	42.5	3.0
Molweni-clean	50	8.5	91.1	3.2

Table 1: Key statistics of corpora: number of documents (#Doc), averaged speech turns, tokens, and speakers per document (#Turn/doc, #Tok/doc, #Spk/doc).

sion), Vicuna, and Mistral in a zero-shot setting.

Implementation Details We use the Vicuna-13b and Mistral-7b models from the Hugging-Face library (Wolf et al., 2020). We employ the *lm-evaluation-harness*[2] framework for computing perplexity scores. We replace speaker names with markers (e.g., John → "spk1") to match the inference setup in the employed models.

5 Results and Analysis

5.1 DS-DP and DS-FLOW Performance

Table 2 shows the performance of the DS-DP and DS-FLOW algorithms on the STAC and Molweni-clean corpora. Precisely, the results for each model include the vanilla version, as well as versions incorporating the penalization factor ($P_{dist}(d)$) and the speech turn limitation (STL). Generally, algorithms utilizing vanilla models perform worse compared to those with constraints; however, they show potential in predicting distant links, as discussed in the following Section 5.3. Applying the STL constraint consistently enhances performance across all metrics. For instance, DS-FLOW on STAC shows an increase in the micro-F_1 score for Vicuna (from 47.2% to 47.7%) and Mistral (from 46.2% to 46.7%). Similarly, DS-DP on STAC improves for Vicuna (from 54.3% to 54.4%) and Mistral (from 53.8% to 54.8%). Comparable improvements are observed on Molweni-clean. These findings suggest that while the STL constraint yields marginal improvements, it reduces complexity by limiting the analysis to fewer sub-dialogues, facilitating a cohesive sub-dialogue examination. Despite predicting complex links with vanilla LLMs and the STL constraint, temporal disparity lowers precision scores (see Section 3.4). When applying the penalization factor $P_{dist}(d)$, significant improvements are noted, as shown in the third row of each group in Table 2. The factor $P_{dist}(d)$ improves results by discouraging longer-distance links and favoring shorter ones, which are more prevalent, as discussed in Section 5.3. Consequently, the best performance on STAC is achieved with the DS-FLOW algorithm using STL and $P_{dist}(d)$. Similarly, the optimal performance on Molweni-clean is obtained with the DS-DP algorithm using STL and $P_{dist}(d)$.

Leveraging the dynamic programming strategy, DS-DP analyzes a larger number of sub-dialogues compared to the greedy approach employed by DS-FLOW, tending to select more short links. This is highlighted by the best performance on Molweni-clean, which involves fewer long-distance links compared to STAC. Conversely, DS-FLOW better predicts longer-distance links, achieving the best performance on the STAC corpus. Overall, when comparing average micro-F_1 scores of DS-DP and DS-FLOW under optimal settings across both corpora, DS-DP slightly outperforms DS-FLOW with scores of 66% versus 65.5%, respectively[3].

Regarding backbone LLMs, Vicuna consistently outperforms Mistral across all settings, highlighting the advantage of models fine-tuned on conversational data for dialogue analysis tasks. However, Mistral demonstrates satisfactory performance, validating the efficacy of the proposed algorithms.

5.2 Unsupervised Method Comparison

We compare our top-performing DS-DP and DS-FLOW settings with other unsupervised methods. Precisely, we consider the following benchmarks[4]: (i) LAST baseline predicts local attachments between adjacent EDUs. Despite its high performance on STAC and Molweni (Muller et al., 2012), it can only extract a single sub-dialogue and cannot detect sub-dialogue structures like our method. (ii) The unsupervised method by Li et al. (2023), which extracts dependency trees from BART model attention matrices (Lewis et al., 2020), fine-tuned through the Sentence Ordering (SO) task. (iii) ChatGPT in a zero-shot setting with a novel prompt for multi-party dialogue discourse parsing, achieving a micro-F_1 score of 52% on STAC, significantly improving over the 20.5% reported by Chan et al. (2023). The prompt is detailed in Appendix H. (iv) Vicuna-13b and Mistral-7b models, prompted identically to ChatGPT in a zero-shot setting.

Table 3 shows the comparison results. Using Vicuna-13b, the DS-DP and DS-FLOW algorithms excel on the STAC corpus, achieving micro-F_1

[2]https://github.com/EleutherAI/lm-evaluation-harness

[3]Qualitative analysis of generated structures is presented in Appendix G.

[4]See Appendix D for additional results pertaining to a smaller Vicuna model.

Model	Algorithm	STAC			Molweni-clean		
		F_1	P	R	F_1	P	R
Vicuna-13b	DS-DP	54.3	52.9	55.8	71.5	68.0	75.3
	DS-DP + STL	54.4	53.3	55.7	72.0	68.9	75.3
	DS-DP + STL + $P_{dist}(d)$	57.3	55.1	59.7	**74.7**	**70.1**	**79.9**
	DS-FLOW	47.2	40.0	57.4	58.1	48.1	73.2
	DS-FLOW + STL	47.7	40.0	59.0	59.3	49.3	74.5
	DS-FLOW + STL + $P_{dist}(d)$	**58.1**	**57.1**	59.2	72.9	69.1	77.2
Mistral-7b	DS-DP	53.8	52.2	55.5	71.1	68.8	73.7
	DS-DP + STL	54.8	53.0	56.6	71.5	68.6	74.5
	DS-DP + STL + $P_{dist}(d)$	56.7	53.4	**60.4**	74.1	69.1	**79.9**
	DS-FLOW	46.2	39.3	55.9	57.0	49.0	68.1
	DS-FLOW + STL	46.7	39.5	57.3	57.3	49.5	68.1
	DS-FLOW + STL + $P_{dist}(d)$	57.0	56.5	57.4	71.0	66.5	76.1

Table 2: Experiment results of proposed approaches on STAC and Molweni-clean corpora. STL: Speech turn limitation. $P_{dist}(d)$: Penalization factor. F_1: Micro-F_1. P: Precision. R: Recall.

Corpus	Baseline	PLM	ChatGPT	Vicuna-13b			Mistral-7b		
	LAST	BART-SO	ZS	ZS	DS-DP	DS-FLOW	ZS	DS-DP	DS-FLOW
STAC	56.8	57.2	52.0	22.8	57.3	**58.1**	30.2	56.7	57.0
Molweni-clean	76.9	-	65.6	35.2	**74.7**	72.9	36.7	74.1	71.0

Table 3: Micro-F_1 scores on STAC and Molweni-clean for the LAST baseline, unsupervised PLM, LLMs within a zero-shot (ZS) setting, and proposed approaches.

scores of 57.3% and 58.1%, respectively, surpassing LAST baseline and BART-SO model. It is noteworthy that the BART-SO model is previously fine-tuned with the SO task on STAC. When employing a vanilla BART model, the performance decreases to 56.6%, representing a 2.6% lower result compared to our method. In comparison, our solution does not require domain-specific data or a fine-tuning process, rendering it easily adaptable to any scenarios. Using Mistral-7b, DS-FLOW outperforms LAST but not BART-SO. On the Molweni-clean corpus, DS-DP and DS-FLOW algorithms lag behind LAST, which achieves 76.9% due to a larger amount of adjacent links in the corpus. Even the strategy proposed by Li et al. (2024a), involving cross-domain training on STAC, only attains a micro-F_1 score of 75.6% on the Molweni-clean corpus, thus trailing behind the LAST baseline. Consequently, a micro-F_1 score of 74.7% (for the DS-DP algorithm incorporating the Vicuna model) may be deemed satisfactory in a fully unsupervised setting. Owing to reproducibility challenges encountered with the BART-SO model on the Molweni-clean corpus, a comparative analysis with our algorithms is not feasible. Finally, in zero-shot settings, Vicuna and Mistral perform abysmally (from -47% to -61% compared to DS-DP and DS-FLOW). ChatGPT outperforms both open-source models, while still falling behind our proposed unsupervised algorithms. Mistral excels at generating structured responses, while Vicuna struggles with lengthy dialogues but outperforms Mistral in our algorithms on both corpora.

5.3 Link Length Analysis

The LAST baseline's limitation is its inability to predict indirect links. To assess the accuracy of our algorithms in predicting distant links, we investigate the performance concerning different link lengths. Figure 3 shows recall scores for different link lengths for DS-FLOW and DS-DP using Vicuna on STAC and Molweni-clean, respectively. We test different settings, including vanilla Vicuna and STL individually for both algorithms. For DS-FLOW on STAC, using vanilla Vicuna accurately predicts long-distance links up to distances of 12 and 13 but increases false positives, as discussed in Section 5.1, affecting precision. Adding STL (DS-FLOW+STL) improves performance for shorter links (distances 1, 2, and 3) and predicts long-distance links up to distance 10. Incorporating $P_{dist}(d)$ with STL (DS-FLOW+STL+PF) achieves over 90% recall for direct links and maintains some ability to predict long-distance links, though performance drops for links over distance 6. Long-distance links (≥ 6) are rare in STAC, under 5% of all links. For DS-DP on Molweni-clean, like DS-FLOW on STAC, both vanilla Vi-

cuna and STL (DS-DP+STL) predict indirect links but not those longer than 4. Including $P_{dist}(d)$ (DS-DP+STL+PF) achieves nearly perfect recall for direct links, with slight performance drops for links at distances 2 and 3 compared to DS-DP+STL. Long-distance links (≥ 4) are rare in Molweni-clean, under 3% of all links. The LAST baseline achieves perfect recall for direct links but fails on long-distance links. In terms of precision and F_1 scores, the STL+PF setting demonstrates higher precision for short-distance links but somewhat lower precision for direct links. All settings experience a decline in F_1 scores as link length increases. An exception is observed for DS-FLOW using the vanilla Vicuna on STAC, which maintains relatively high F_1 scores for links at distances of 12 and 13. Further evaluation results are in Appendix A.

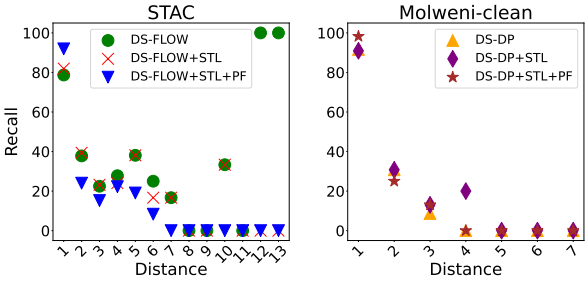

Figure 3: Recall scores for different link lengths. The left plot shows three settings with DS-FLOW on STAC; the right plot depicts the same settings for DS-DP on Molweni-clean. Both algorithms use Vicuna as LLM. STL: Speech turn limitation. STL+PF: Speech turn limitation in combination with penalization factor.

6 Discussion

Despite its significantly lower parameter count compared to the Vicuna-13b model, we used the Mistral-7b model for our algorithms' assessment owing to its superior performance relative to larger models like LLaMa 2-13b (Touvron et al., 2023b) and LLaMa 1-34b (Touvron et al., 2023a) across multiple benchmarks.

Although our algorithms exhibit polynomial complexity, employing exceedingly large models increases the computational time required for calculating perplexity in extended dialogues[5]. Presently, the state-of-the-art lacks alternative unsupervised metrics with efficient time complexity for evaluating dialogue quality. Metrics such as FED (Mehri and Eskénazi, 2020) and FULL (Bruyn et al., 2022) entail computing multiple log-likelihood values for dialogue assessment, contrasting with perplexity, which mandates the computation of a singular log-likelihood value[6]. We leave the comparison among these metrics for future investigations.

7 Conclusion and Future Work

In this paper, we introduce an innovative, fully unsupervised method for extracting discourse structures in multi-party dialogues. To this end, we leverage open-source LLMs and introduce two algorithms, DS-DP and DS-FLOW, to detect coherent sub-dialogues within a dialogue. On the STAC and Molweni-clean corpora, we achieve micro-F_1 scores of 58.1% and 74.7%, respectively, demonstrating the efficacy of our solution in constructing dialogue structures without the need for labeled data. In the future, we intend to enhance the coherence evaluation metric, particularly addressing cognitive aspects as in Li et al. (2024b), and explore applying LLMs for unsupervised prediction of rhetorical relation types to deduce full discourse structures. Furthermore, we aim to improve algorithm link selection by incorporating linguistically motivated constraints as in Perret et al. (2016). Lastly, we plan to evaluate our architectural choices across diverse corpora and discourse parsing tasks to further validate their efficacy in assessing dialogues in real-world scenarios.

Acknowledgments

The authors thank the anonymous reviewers for their insightful comments and suggestions. This research was supported by Mitacs as part of the Globalink Research Award (GRA) program. We acknowledge their assistance and funding, which made this study possible.

References

Stergos Afantenos, Nicholas Asher, Farah Benamara, Anais Cadilhac, Cedric Dégremont, Pascal Denis, Markus Guhe, Simon Keizer, Alex Lascarides, Oliver Lemon, et al. 2012. Modelling strategic conversation: model, annotation design and corpus. In *Proceedings of SemDial 2012 (SeineDial): The 16th Workshop on the Semantics and Pragmatics of Dialogue*, pages 167–168.

Nicholas Asher, Julie Hunter, Mathieu Morey, Farah Benamara, and Stergos D. Afantenos. 2016. Discourse structure and dialogue acts in multiparty dialogue:

[5]Detailed insights into the algorithm execution times are provided in Appendix F.

[6]An examination of the constraints associated with perplexity can be found in Appendix G.

the STAC corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

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

Sonia Badene, Kate Thompson, Jean-Pierre Lorré, and Nicholas Asher. 2019a. Data programming for learning discourse structure. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 640–645. Association for Computational Linguistics.

Sonia Badene, Kate Thompson, Jean-Pierre Lorré, and Nicholas Asher. 2019b. Weak supervision for learning discourse structure. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2296–2305. Association for Computational Linguistics.

Zineb Bennis, Julie Hunter, and Nicholas Asher. 2023. A simple but effective model for attachment in discourse parsing with multi-task learning for relation labeling. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 3404–3409. Association for Computational Linguistics.

Maxime De Bruyn, Ehsan Lotfi, Jeska Buhmann, and Walter Daelemans. 2022. Open-domain dialog evaluation using follow-ups likelihood. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 496–504. International Committee on Computational Linguistics.

Chunkit Chan, Jiayang Cheng, Weiqi Wang, Yuxin Jiang, Tianqing Fang, Xin Liu, and Yangqiu Song. 2023. Chatgpt evaluation on sentence level relations: A focus on temporal, causal, and discourse relations. *CoRR*, abs/2304.14827.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Davide Colla, Matteo Delsanto, Marco Agosto, Benedetto Vitiello, and Daniele Paolo Radicioni. 2022. Semantic coherence markers: The contribution of perplexity metrics. *Artif. Intell. Medicine*, 134:102393.

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 2022, Vienna, Austria, 23-29 July 2022*, pages 5453–5460. ijcai.org.

Raquel Fernández, Matthew Frampton, Patrick Ehlen, Matthew Purver, and Stanley Peters. 2008. Modelling and detecting decisions in multi-party dialogue. In *Proceedings of the SIGDIAL 2008 Workshop, The 9th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 19-20 June 2008, Ohio State University, Columbus, Ohio, USA*, pages 156–163. The Association for Computer Linguistics.

Matthew Frampton, Jia Huang, Trung H. Bui, and Stanley Peters. 2009. Real-time decision detection in multi-party dialogue. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1133–1141. ACL.

Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey Bigham. 2022. InstructDial: Improving zero and few-shot generalization in dialogue through instruction tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 505–525, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuchen He, Zhuosheng Zhang, and Hai Zhao. 2021. Multi-tasking dialogue comprehension with discourse parsing. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation, PACLIC 2021, Shanghai International Studies University, Shanghai, China, 5-7 November 2021*, pages 551–561. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Shafiq R. Joty, Giuseppe Carenini, Raymond T. Ng, and Gabriel Murray. 2019. Discourse analysis and its applications. In *Proceedings of the 57th Conference of the Association for Computational Linguistics: Tutorial Abstracts, ACL 2019, Florence, Italy, July 28, 2019, Volume 4: Tutorial Abstracts*, pages 12–17. Association for Computational Linguistics.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021. Dialogue state tracking with a language model using schema-driven prompting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, November 7-11, 2021*, pages 4937–4949. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

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 5th Workshop on Computational Approaches to Discourse (CODI 2024)*, pages 161–176.

Chuyuan Li, Patrick Huber, Wen Xiao, Maxime Amblard, Chloé Braud, and Giuseppe Carenini. 2023. Discourse structure extraction from pre-trained and fine-tuned language models in dialogues. In *Findings of the Association for Computational Linguistics: EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 2517–2534. 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, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 2642–2652. International Committee on Computational Linguistics.

Jiaqi Li, Ming Liu, Bing Qin, and Ting Liu. 2022. A survey of discourse parsing. *Frontiers Comput. Sci.*, 16(5):165329.

Manling Li, Lingyu Zhang, Heng Ji, and Richard J. Radke. 2019. Keep meeting summaries on topic: Abstractive multi-modal meeting summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2190–2196. Association for Computational Linguistics.

Xue Li, Jia Su, Yang Yang, Zipeng Gao, Xinyu Duan, and Yi Guan. 2024b. Dialogues are not just text: Modeling cognition for dialogue coherence evaluation. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 18573–18581. AAAI Press.

Annie Louis, Aravind Joshi, and Ani Nenkova. 2010. Discourse indicators for content selection in summarization. In *Proceedings of the SIGDIAL 2010 Conference*, pages 147–156.

Ryan Lowe, Nissan Pow, Iulian 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 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2-4 September 2015, Prague, Czech Republic*, pages 285–294. The Association for Computer Linguistics.

Tiezheng Mao, Tianyong Hao, Jialing Fu, and Osamu Yoshie. 2024. Hierarchical graph fusion network and a new argumentative dataset for multiparty dialogue discourse parsing. *Inf. Process. Manag.*, 61(2):103613.

Shikib Mehri and Maxine Eskénazi. 2020. Unsupervised evaluation of interactive dialog with dialogpt. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGdial 2020, 1st virtual meeting, July 1-3, 2020*, pages 225–235. Association for Computational Linguistics.

Philippe Muller, Stergos D. Afantenos, Pascal Denis, and Nicholas Asher. 2012. Constrained decoding for text-level discourse parsing. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012), December 8-15, 2012, Mumbai, India*, pages 1883–1900. Indian Institute of Technology Bombay.

Noriki Nishida and Yuji Matsumoto. 2022. Out-of-domain discourse dependency parsing via bootstrapping: An empirical analysis on its effectiveness and limitation. *Trans. Assoc. Comput. Linguistics*, 10:127–144.

Jérémy Perret, Stergos D. Afantenos, Nicholas Asher, and Mathieu Morey. 2016. Integer linear programming for discourse parsing. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016): Human Language Technologies (HLT) , San Diego California, USA, June 12-17, 2016*, pages 99–109. Association for Computational Linguistics.

Vipula Rawte, Amit P. Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *CoRR*, abs/2309.05922.

Emanuel A Schegloff. 2007. *Sequence organization in interaction: A primer in conversation analysis I*, volume 1. Cambridge university press.

Kai Sun, Su Zhu, Lu Chen, Siqiu Yao, Xueyang Wu, and Kai Yu. 2016. Hybrid dialogue state tracking for real world human-to-human dialogues. In *Proceedings of the 17th Annual Conference of the International Speech Communication Association (Interspeech 2016), San Francisco, CA, USA, September 8-12, 2016*, pages 2060–2064. ISCA.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal

Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/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 30th International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 3943–3949. ijcai.org.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Wen Xiao, Patrick Huber, and Giuseppe Carenini. 2020. Do we really need that many parameters in transformer for extractive summarization? discourse can help ! *CoRR*, abs/2012.02144.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5021–5031. Association for Computational Linguistics.

Chen Zhang, Luis Fernando D'Haro, Yiming Chen, Malu Zhang, and Haizhou Li. 2024. A comprehensive analysis of the effectiveness of large language models as automatic dialogue evaluators. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI 2024), 36th Conference on Innovative Applications of Artificial Intelligence (IAAI 2024), 14th Symposium on Educational Advances in Artificial Intelligence (EAAI 2014), February 20-27, 2024, Vancouver, Canada*, pages 19515–19524. AAAI Press.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023. Instruction tuning for large language models: A survey. *CoRR*, abs/2308.10792.

A Further Evaluation Results for Link Length Analysis

To further evaluate our approaches, we analyze the precision and F_1 scores for link lengths using DS-FLOW and DS-DP with Vicuna on the STAC and Molweni-clean corpora. As in Section 5.3, we explore different settings. Figure 4 shows that the STL+PF setting, which includes both the STL constraint and the penalization factor, provides the best precision scores for links with distances ranging from 2 to 5 in STAC and from 2 to 3 in Molweni-clean. This setting improves the evaluation of short-distance links, resulting in fewer false positives, but slightly lower precision ($\sim 3\%$) for direct links due to the penalization factor. Additionally, although STL and vanilla settings predict long-distance links, they introduce several false positives. For instance, DS-FLOW with vanilla Vicuna on STAC predicted incorrect links with distances ranging from 14 to 31. Figure 5 highlights that in both corpora, all settings show decreasing F_1 scores as link lengths increase, except for DS-FLOW with vanilla Vicuna on STAC, which achieves an F_1 score of 40% for links of length 13 and 13% for links of length 12.

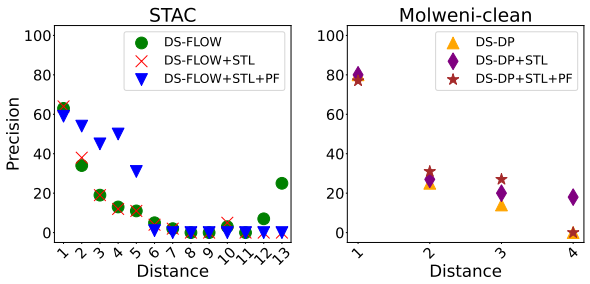

Figure 4: Precision scores for different link lengths. The left plot shows three settings with DS-FLOW on STAC; the right plot depicts the same settings for DS-DP on Molweni-clean. Both algorithms use Vicuna as LLM. STL: Speech turn limitation. STL+PF: Speech turn limitation in combination with penalization factor.

B Filtering Mechanism Prompt Template

Drawing inspiration from the prompt utilized by Zhang et al. (2024) for evaluating dialogue qualities, we devise a new prompt specifically tailored to the task of predicting potential dialogue continuations, as depicted in Figure 6. We adapt the instruction template to align with the format used by Vicuna and Mistral in their instruction-tuning process.

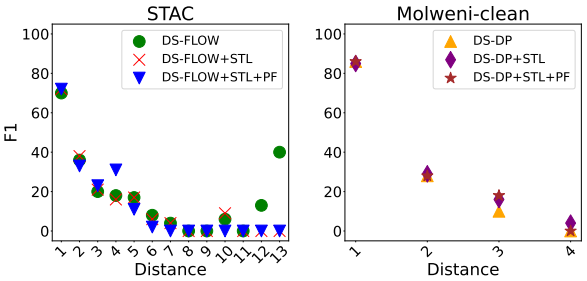

Figure 5: F_1 scores for different link lengths. The left plot shows three settings with DS-FLOW on STAC; the right plot depicts the same settings for DS-DP on Molweni-clean. Both algorithms use Vicuna as LLM. STL: Speech turn limitation. STL+PF: Speech turn limitation in combination with penalization factor.

Context:
[Here is a dialogue utterance]
Response:
[Here is the potential continuation]
Instruction:
Please evaluate whether the response is a plausible continuation of the given utterance within a dialogue context and provide a definitive answer Yes or No.
Your Answer:
[Here is LLM's output in terms of "Yes" or "No"]

Figure 6: An example of how open-source LLMs can be prompted to determine if an utterance could potentially follow a preceding one.

C Filtering Mechanism Evaluation

We assessed the performance of the filtering mechanism under the optimal setting for the DS-FLOW algorithm, specifically leveraging the STL constraint, the penalization factor, and the Vicuna-13b model as the backbone on the STAC corpus. The following scenarios were considered:

- **True Positives**: Links that should be filtered and are correctly identified as such by the LLM (112 instances).
- **False Positives**: Links that should not be filtered but are incorrectly identified as filtered by the LLM (52 instances).
- **True Negatives**: Links that should not be filtered and are correctly identified as such by the LLM (698 instances).
- **False Negatives**: Links that should be filtered but are incorrectly identified as not filtered by the LLM (604 instances).

Model	Algorithm	STAC			Molweni-clean		
		F_1	P	R	F_1	P	R
Vicuna-7b	DS-DP	53.8	51.7	56.0	69.6	66.4	73.2
	DS-DP + STL	54.4	52.3	56.6	69.9	66.7	73.5
	DS-DP + STL + $P_{dist}(d)$	56.9	53.2	61.3	73.2	67.6	79.9
	DS-FLOW	46.0	38.6	56.9	57.6	47.7	72.7
	DS-FLOW + STL	46.4	39.3	56.6	58.5	48.7	73.2
	DS-FLOW + STL + $P_{dist}(d)$	56.3	55.1	57.5	72.8	68.4	77.7

Table 4: Experiment results of proposed approaches on STAC and Molweni-clean corpora for Vicuna-7b. STL: Speech turn limitation. $P_{dist}(d)$: Penalization factor. F_1: Micro-F_1. P: Precision. R: Recall.

Based on these outcomes, we calculated the Accuracy, Precision, Recall, and F_1 scores, as detailed in Table 5. The filtering mechanism exhibits good reliability, demonstrated by a Precision score of 68.2%. However, it only filters out a small number of incorrect potential continuations, resulting in a Recall score of 15.6%, which in turn affects the F_1 score. The overall Accuracy score of 55.3% is consistent with the algorithm's performance on the STAC corpus. Enhancing the filtering mechanism is expected to improve the algorithm's performance, a subject we plan to address in future work.

Metric	**Value (%)**
Precision	68.2
Recall	15.6
Accuracy	55.3
F_1	25.1

Table 5: Performance metrics of the filtering mechanism under the optimal setting for the DS-FLOW algorithm, leveraging the STL constraint, the penalization factor, and the Vicuna-13b model on the STAC corpus.

D Experimental Analysis with Smaller Vicuna Model

To analyze how performance changes with LLM model size, we conduct supplementary analyses using a smaller version of Vicuna, comprising 7b parameters. As shown in Table 4, akin to Vicuna-13b (see Table 2), both algorithms exhibit optimal performance when incorporating both the STL constraint and the penalization factor, with a slight improvement observed when integrating the STL constraint compared to the vanilla version. Regarding the best settings, the results indicate that with the downsized LLM, the micro-F_1 scores are slightly lower, with DS-FLOW achieving 56.3% and DS-DP achieving 73.2%, compared to Vicuna-13b, which achieved 58.1% on STAC and 74.7% on Molweni-clean, respectively. This suggests that employing a larger LLM could potentially yield superior outcomes.

E An Example of DS-DP Algorithm Execution

Figure 7 illustrates the application of DS-DP to the dialogue depicted in Figure 1. The algorithm begins by calculating perplexity scores for sub-dialogues of length 2 during the initialization phase. It then progresses to compute the perplexity scores for sub-dialogues of length 3. Specifically, when $k = 1$, the algorithm analyzes all pairs of EDUs (e_i, e_j) with $i > 1$ and $j > i$. Given the constraint that sub-dialogues must start from the initial EDU (see Section 3.1), every pair of sequential EDUs has the initial EDU as the preceding one.

For $k = 2$, the algorithm computes sub-dialogues of length 4. For the cells $M_{\text{co}}[3][4][2]$, $M_{\text{co}}[3][5][2]$, and $M_{\text{co}}[3][6][2]$, there is only one possible sub-dialogue, and the algorithm computes their perplexity scores. When evaluating the sub-dialogue passing through e_4 and ending in e_5, the algorithm analyzes the incoming links in e_4. According to the input graph, e_4 has incoming links from e_1, e_2, and e_3. Since there is no sub-dialogue passing through e_1, ending in e_4, and involving $k - 1 = 1$ EDU, e_1 is disregarded. Instead, the algorithm considers e_2 and e_3 as intermediary EDUs to conclude in e_5 through e_4. Thus, it analyzes two sub-dialogues (A): (e_1, e_2, e_4, e_5) and (e_1, e_3, e_4, e_5). The algorithm computes the perplexity scores for both sub-dialogues and retains the one with the lowest perplexity, e.g., (e_1, e_3, e_4, e_5). The same method applies to sub-dialogues passing through e_4 and ending in e_6 (B), and those passing through e_5 and ending in e_6 (C). Consider the selection of sub-dialogues (e_1, e_3, e_4, e_6) and (e_1, e_2, e_5, e_6).

For $k = 3$, the algorithm examines pairs of EDUs that can involve three preceding EDUs, such as (e_4, e_5), (e_4, e_6), and (e_5, e_6). The first two pairs

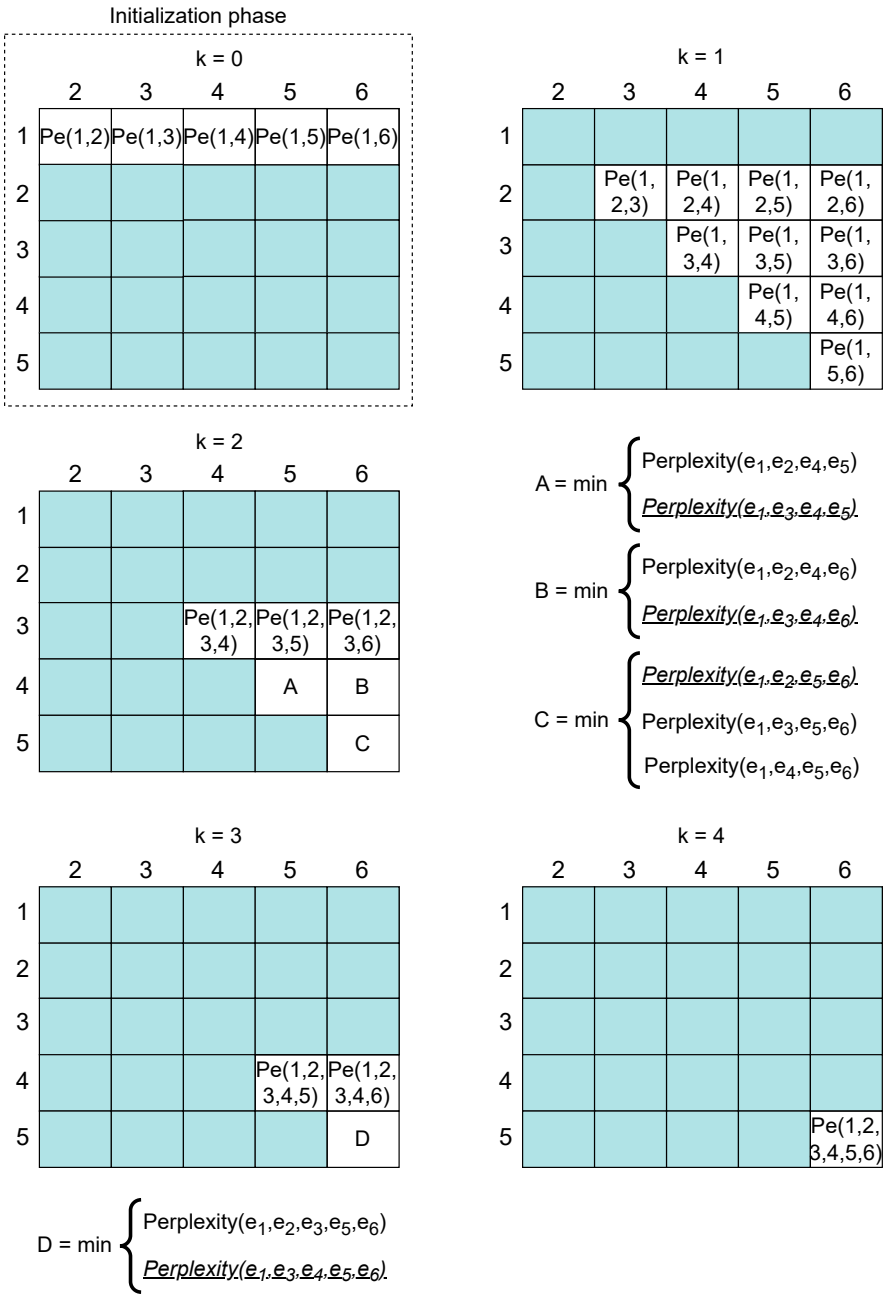

Figure 7: An example of DS-DP execution. For brevity, we use the notation *Pe*(list of indexes) instead of *Perplexity*(list of EDUs) within the cells. Underlined texts denote selected sub-dialogues during the algorithm's execution.

are constrained by preceding EDUs e_1, e_2, and e_3. For the pair (e_5, e_6), multiple triples can serve as the preceding three EDUs. Here, the algorithm considers EDUs with outgoing links towards e_5, namely e_1, e_2, e_3, and e_4. Only two EDUs, e_3 and e_4, can reach e_5 and involve two preceding EDUs. Since the algorithm selected the sub-dialogue (e_1, e_3, e_4, e_5) as the best option for passing through e_4 and ending in e_5 with two preceding EDUs, it does not analyze the sub-dialogue (e_1, e_2, e_4, e_5, e_6) and just considers (e_1, e_3, e_4, e_5, e_6). Conversely, the only sub-dialogue passing through e_3 and ending in e_5 with two preceding EDUs is (e_1, e_2, e_3, e_5), leading the algorithm to analyze (e_1, e_2, e_3, e_5, e_6). Consider the selection of (e_1, e_3, e_4, e_5, e_6).

For $k = 4$, the only pair of EDUs that can have four preceding EDUs is (e_5, e_6), resulting in the sub-dialogue (e_1, e_2, e_3, e_4, e_5, e_6). With this, the algorithm completes the computation of the most coherent sub-dialogues with lengths ranging from 2 to 6. Then, it iterates over the k value for each EDU e_i and selects the sub-dialogue ending in e_i with the minimum perplexity, examining only the column i for each k. For example, to find the most coherent sub-dialogue ending in e_4, it evaluates the perplexity scores of the following sub-dialogues: (e_1, e_4), (e_1, e_2, e_4), (e_1, e_3, e_4), and (e_1, e_2, e_3, e_4). In the context of the dialogue in Figure 1, the algorithm selects (e_1, e_2) for e_2, (e_1, e_3) for e_3, (e_1, e_3, e_4) for e_4, (e_1, e_3, e_4, e_5) for e_5, and (e_1, e_3, e_4, e_6) for e_6, resulting in the final DAG: $\{(e_1, e_2), (e_1, e_3), (e_3, e_4), (e_4, e_5), (e_4, e_6)\}$.

From the matrices, it is evident that there is a significant number of empty cells (indicated in light blue). For each k value, the algorithm only needs to examine rows with indices greater than k, and for each row i, only columns with indices greater than i. This is justified by the assumption of not having backward links within the final DAG to be computed.

F Algorithm Execution Time Analysis

We assessed the execution times of the proposed algorithms using the STAC and Molweni-clean corpora. Although the DS-FLOW algorithm exhibits a time complexity of $O(|V|^2)$, which is more favorable compared to the $O(|V|^4)$ complexity of the DS-DP algorithm, our empirical analysis revealed that the DS-DP algorithm computes discourse structures more efficiently, with execution times sometimes reduced by up to half. This discrepancy occurs because, even though the DS-DP algorithm has a $O(|V|^4)$ time complexity, it processes fewer values than expected in the worst-case scenario (as detailed in Appendix E). Furthermore, the DS-FLOW algorithm requires LLM computation for both filtering and perplexity calculations, while the DS-DP algorithm uses an LLM solely for evaluating sub-dialogue coherence.

G Qualitative Analysis in STAC and Molweni-clean

In Figures 8-19, we present several concrete examples generated by the optimal approaches for STAC (utilizing DS-FLOW with the STL constraint and penalization factor, and Vicuna-13b as the backbone) and Molweni-clean (employing DS-DP with identical settings). Specifically, we show three well-predicted examples (depicted in Figures 8, 9, and, 10 for STAC, and Figures 14, 15, and 16 for Molweni-clean) and three badly predicted examples (depicted in Figures 11, 12, and 13 for STAC, and Figures 17, 18, and 19 for Molweni-clean). Some patterns observed in predicted structures include: (i) the algorithms struggle to predict very long-distance links, favoring shorter links with distances of 2, 3, and 4; (ii) direct links are often predicted even when the appropriate indirect incoming links for EDUs are accurately identified.

Our qualitative analysis has identified multiple instances wherein the application of perplexity for dialogue evaluation presents limitations. To exemplify this issue, consider the following dialogue excerpt from the STAC corpus (id *s1-league3-game3_16*):

(e_1) **A**: can anyone trade ore? I have more wood to trade
(e_2) **B**: do you have clay, by any chance?
(e_3) **A**: sorry, no
(e_4) **C**: i can do that kieran
(e_5) **A**: how many can you trade?
(e_6) **A**: 2 for 2?
(e_7) **C**: just got one, sorry
(e_8) **A**: ok cool

In this instance, our algorithms evaluated the links (e_5, e_7) as more likely than (e_5, e_6), despite both links being valid: (e_5, e_7) with a *QAP* relation and (e_5, e_6) with a *Continuation* relation. This discrepancy is likely because a direct answer like e_7 seems more contextually relevant as a response to e_5, thus overshadowing the *Continuation* link to e_6.

The perplexity metric tends to favor more immediate and clear connections, which can sometimes misrepresent the actual flow of dialogue. This limitation indicates that relying solely on perplexity for dialogue evaluation may overlook nuanced conversational dynamics, underscoring the need for supplementary metrics to fully capture dialogue coherence and relevance.

H Dialogue Parsing Task Prompt Template

To enhance the competitive performance of ChatGPT in the multi-party dialogue discourse parsing task, we undertake manual design efforts to refine the prompt. This refinement, illustrated in Figure 20, builds upon the prompt proposed by Chan et al. (2023). Specifically, we provide a more explicit delineation of the task requirements by specifying the extraction of a DAG, in contrast to the broader objective pursued by Chan et al. (2023), which involved predicting all potential discourse relations between utterances. Furthermore, drawing on insights from the findings of Chan et al. (2023), who demonstrated improved performance with the inclusion of descriptions for discourse relations, we develop more comprehensive descriptions within the prompt. This refined prompt has been consistently used in zero-shot experiments conducted with the Vicuna and Mistral models.

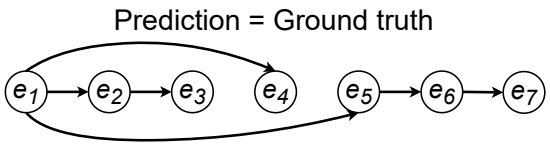

Figure 8: STAC - DS-FLOW - Well predicted example: *pilot02_12*. $\#EDUs$: 7.

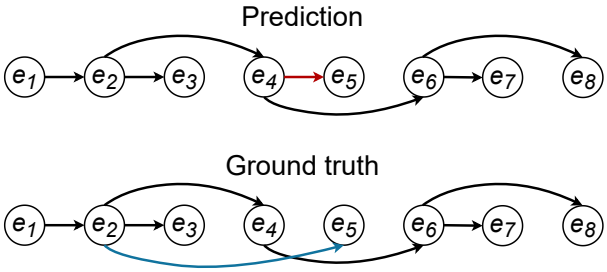

Figure 9: STAC - DS-FLOW - Well predicted example: *pilot02_21*. $\#EDUs$: 8. In red: False positive edges; in light blue: False negative edges.

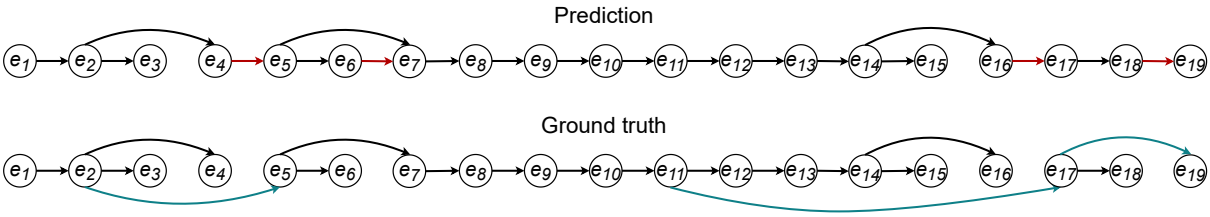

Figure 10: STAC - DS-FLOW - Well predicted example: *pilot02_13*. $\#EDUs$: 19. In red: False positive edges; in light blue: False negative edges.

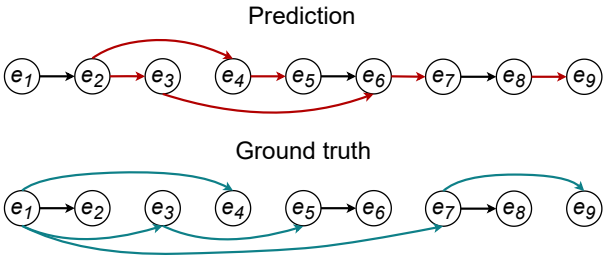

Figure 11: STAC - DS-FLOW - Badly predicted example: *s2-league4-game2_6*. $\#EDUs$: 9. In red: False positive edges; in light blue: False negative edges.

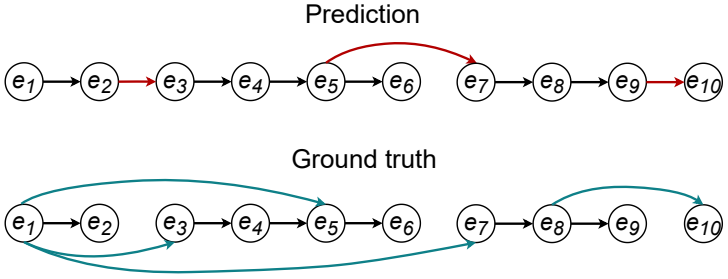

Figure 12: STAC - DS-FLOW - Badly predicted example: *pilot02_6*. $\#EDUs$: 10. In red: False positive edges; in light blue: False negative edges.

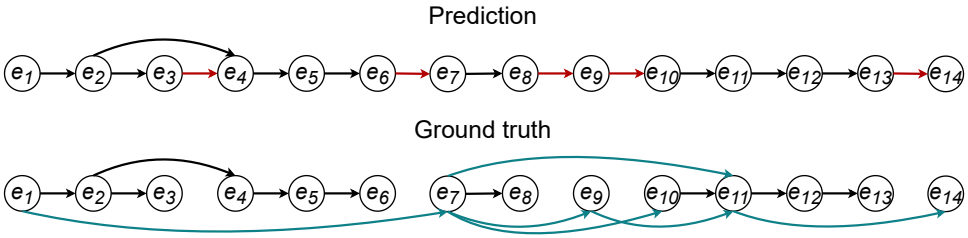

Figure 13: STAC - DS-FLOW - Badly predicted example: *s2-league4-game2_31*. $\#EDUs$: 14. In red: False positive edges; in light blue: False negative edges.

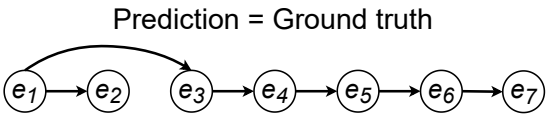

Figure 14: Molweni-clean - DS-DP - Well predicted example: *8031*. $\#EDUs$: 7.

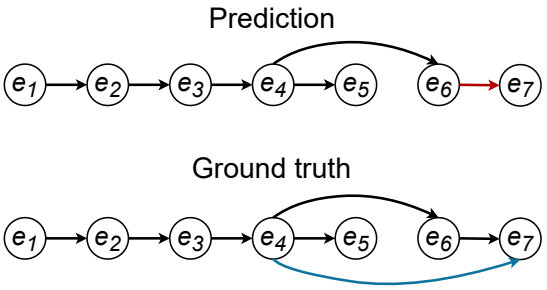

Figure 15: Molweni-clean - DS-DP - Well predicted example: *6037*. $\#EDUs$: 7. In red: False positive edges; in light blue: False negative edges.

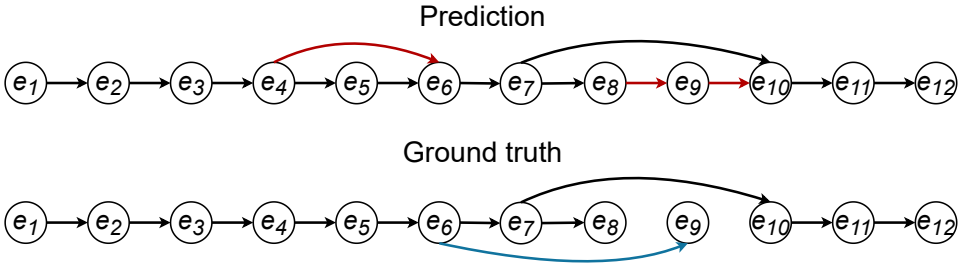

Figure 16: Molweni-clean - DS-DP - Well predicted example: *8026*. $\#EDUs$: 12. In red: False positive edges; in light blue: False negative edges.

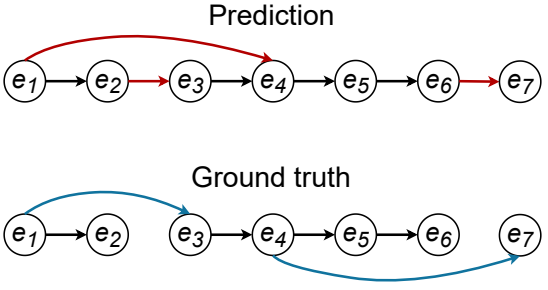

Figure 17: Molweni-clean - DS-DP - Badly predicted example: *5033*. $\#EDUs$: 7. In red: False positive edges; in light blue: False negative edges.

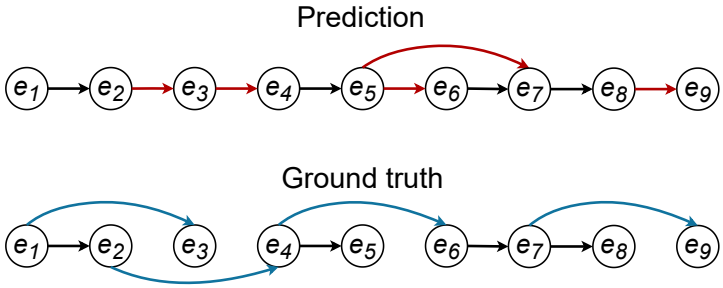

Figure 18: Molweni-clean - DS-DP - Badly predicted example: *8039*. $\#EDUs$: 9. In red: False positive edges; in light blue: False negative edges.

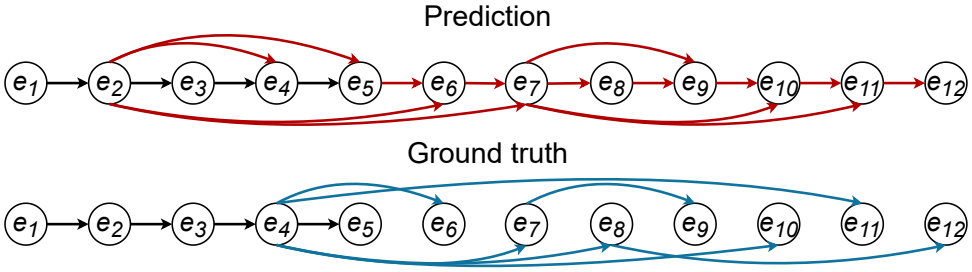

Figure 19: Molweni-clean - DS-DP - Badly predicted example: *8018*. $\#EDUs$: 12. In red: False positive edges; in light blue: False negative edges.

Here is a multi-party dialogue:
[Multi-party dialogue]
Assume that each utterance represents a node within a graph. Your task is to predict the relations between these utterances based on a provided list of relations. The resulting graph should adhere to the structure of a Directed Acyclic Graph (DAG), wherein edges have a direction, meaning they go from one node to another. A key characteristic of a DAG is that it does not contain cycles, i.e., there are no sequences of edges that form a closed loop. This implies that it is not possible to start from a node, follow the edges, and return to the starting node. It is crucial to emphasize that each node representing an utterance must have at least one incoming edge to ensure that the resulting graph maintains coherence and fosters a connected discourse.

Relations:
1) *Comment:* This relation type typically indicates that one utterance provides a comment or opinion on the content of another utterance. It shows a speaker's perspective or evaluation of the preceding statement.
2) *Clarification Question:* In this relation, one utterance poses a question seeking clarification or additional information about the content of another utterance. It implies a request for further explanation.
3) *Elaboration:* Elaboration signifies that one utterance expands upon or provides more details about the content of another utterance. It is used to enhance understanding by offering additional information or context.
4) *Continuation:* This relation suggests that one utterance continues the topic or discussion from a previous utterance. It signifies a logical progression in the conversation.
5) *Explanation:* Explanation pertains to one utterance offering an explanation or clarification in response to a question or confusion expressed in another utterance. It aids in providing clarity.
6) *Conditional:* A conditional relation implies that one utterance presents a condition or hypothetical scenario related to the content of another utterance. It often involves "if-then" statements.
7) *Question-Answer Pair:* This relation indicates that one utterance contains a question, and another utterance follows with an answer to that question. It demonstrates a direct question-and-answer interaction.
8) *Alternation:* Alternation shows that two utterances present alternative options or choices. It is used when discussing multiple possibilities or courses of action.
9) *Q-Elab:* Q-Elab signifies that one utterance asks a question, and another utterance follows with an elaboration or further explanation of the question or its context.
10) *Result:* Result indicates that one utterance discusses the outcome or consequence of the content presented in another utterance. It shows a cause-and-effect relation.
11) *Background:* In this relation, one utterance provides background information or context that is relevant to the content of another utterance. It helps set the stage for the discussion.
12) *Narration:* Narration signifies that one utterance presents a narrative or storytelling element, often in response to a question or to share an anecdote.
13) *Correction:* Correction shows that one utterance corrects or revises the content of another utterance. It is used to rectify errors or inaccuracies.
14) *Parallel:* Parallel relations occur when two or more utterances share similar or related content, often in a parallel or analogous manner. It emphasizes similarities or comparisons.
15) *Contrast:* Contrast signifies that one utterance presents content that is in contrast or opposition to the content of another utterance. It highlights differences or contradictions in the conversation.

Figure 20: Prompt template employed for LLMs in a zero-shot setting for the multi-party dialogue discourse parsing task on the STAC and Molweni-clean corpora.