Dialogue Discourse Parsing as Generation: a Sequence-to-Sequence LLM-based Approach

Chuyuan Li, Yuwei Yin, Giuseppe Carenini
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada, V6T 1Z4
chuyuan.li@ubc.ca; {yuweiyin, carenini}@cs.ubc.ca

Abstract

Discourse analysis studies the sentence organization within a document, aiming to reveal its underlying structural information. Existing works on dialogue discourse parsing mostly use encoder-only models and sophisticated decoding strategies to extract structures. Despite recent advances in Large Language Models (LLMs), applying directly these models on discourse parsing is challenging. To fully leverage the rich semantic and discourse knowledge in LLMs, we propose to transform discourse parsing into a generation task using a text-to-text paradigm. Our approach is intuitive and requires no modification of the LLM architecture. Experimental results on STAC and Molweni datasets show that a sequence-to-sequence model such as T0 can perform reasonably well. Notably, our improved transition-based sequence-to-sequence system achieves new state-of-the-art performance on Molweni. Furthermore, our systems can generate richer discourse structures such as graphs, whereas previous methods are mostly limited to trees.[1]

1 Introduction

Discourse parsing is a Natural Language Processing task that aims to retrieve a structure from a document. The discursive structure contains clause-like text spans (known as Elementary Discourse Units) and are linked by semantic-pragmatic relations such as *Elaboration* and *Acknowledgment*. It plays a crucial role in natural language understanding and has demonstrated its usefulness in various downstream applications such as summarization (Feng et al., 2021) and dialogue comprehension (He et al., 2021; Ma et al., 2023).

Existing works on Dialogue Discourse Parsing (DDP) suggest that task-specific models are necessary to achieve state-of-the-art (SOTA) performance (Chi and Rudnicky, 2022; Li et al., 2023a). They are based on complex architectures constructed on top of encoder-only pre-trained language models (PLMs) such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). These models present a few limitations. First, they require task-specific architectures which oftentimes involve heavy engineering of utterance embeddings and specialized decoding strategies. Second, the predicted structures are typically limited to *trees*, neglecting other rich representations such as directed acyclic graphs (Asher et al., 2016). Third, they do not leverage rich latent knowledge in more recent Large decoder-only and encoder-decoder Language Models (LLMs) (Brown et al., 2020; Sanh et al., 2022; Chowdhery et al., 2023; Touvron et al., 2023).

Such LLMs have shown remarkable abilities in a wide range of applications, from text understanding and generation to coding to reasoning (Bang et al., 2023; Bubeck et al., 2023), resulting in a shift in focus from relatively small encoder-only PLMs to large-scale encoder-decoder and decoder-only LLMs. LLMs see a great amount of data: T0 model (Sanh et al., 2022), for instance, is pretrained on the C4 corpus (Habernal et al., 2016) containing 356 billion tokens; they are pretrained on a mixture of downstream tasks such as multi-document question answering (Yang et al., 2018) and natural language inference (Bowman et al., 2015). Since many of these tasks require an understanding of the inter-sentence structure, we hypothesize that LLMs have good contextual representation for sentence-level reasoning (e.g., discourse analysis).

However, in our preliminary experiments, we found that directly prompting LLMs does not perform well on the DDP task, confirming similar observations by Chan et al. (2023) who applied zero-shot prompting and in-context learning methods but found poor performance with GPT-3.5.

In this paper, we ask the question: *how to effectively transform the discourse parsing task into a*

[1]Code is available at https://github.com/chuyuanli/Seq2Seq-DDP.

Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 1–14
September 18–20, 2024.

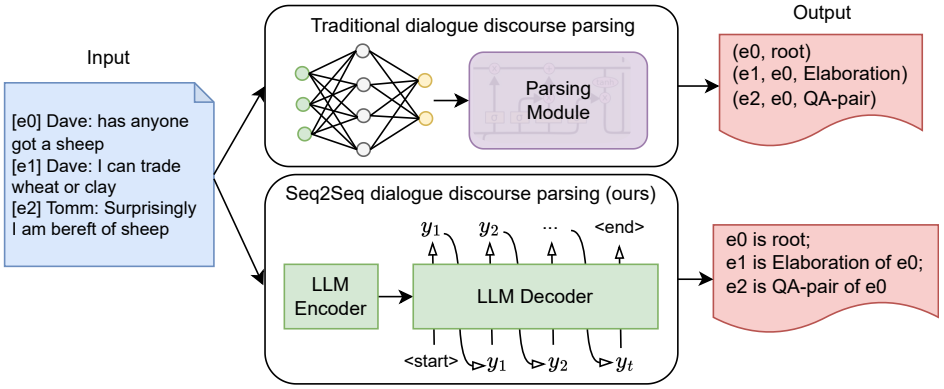

Figure 1: Traditional dialogue discourse parsing and our Seq2Seq dialogue discourse parsing systems. e_i denotes the discourse units and "QA-pair" represents the question-answer pair.

generation task?

To this end, we propose to tackle this problem within a text-to-text paradigm. We first formalize the parsing task as a Seq2Seq process and present a simple approach where a model takes a sequence of raw texts as input and produces a sequence of structures as output. We call this system **Seq2Seq-DDP**. The adopted model, such as T0, has a standard encoder-decoder architecture and is fine-tuned on parsing task. There is a great deal of flexibility in sequence representations, especially for the target sequence where *tree-like* and *graph-like* structures need to be expressed linearly. We design different schemes: one is close to natural language and another one is inspired by the *translation between augmented natural language* (TANL) formats (Paolini et al., 2021). This approach is straightforward, but it is constrained by weak supervision with lengthy inputs, which can lead to hallucinated or missing predictions for some utterances.

To tackle these issues, we propose to improve our system with transition-based algorithms which are widely used in dependency parsing (Nivre, 2003, 2008). A transition-based model receives the states of parsed sentences and the target sentence and predicts an action corresponding to the target sentence. A recent work on coreference resolution implemented such a system and achieved SOTA performance (Bohnet et al., 2023). Our enhanced system, **Seq2Seq-DDP+Transition**, processes one sentence at each step and predicts an action that establishes links and relations towards that sentence. We also adapt the sequence representations accordingly. Compared to the previous approach using full text input and output, the new system is more controllable with partial inputs and outputs.

We evaluate both systems on the STAC and Molweni datasets. The Seq2Seq-DDP model delivers promising results, matching the performance of SOTA models on Molweni. The transition-based system provides significant improvements across both datasets, setting new SOTA on Molweni. Through a series of analyses, we identify several key factors in converting a parsing task into a generation task, including the amount of supervision and the design of the representation scheme.

To summarize: (1) we propose a Seq2Seq-DDP method, along with an improved Seq2Seq-DDP+Transition variant, to transform discourse parsing into an LLM-based generation task, where our sophisticated sequence representations deliver promising performance gains; (2) we conduct extensive experiments and comprehensive analyses, which reveal insightful ideas on what makes a successful generative model for discourse parsing.

2 Related Work

Discourse Parsing Discourse parsing is a hard task, with low performance especially for multi-party dialogues which involve intricate relations between speakers, such as STAC (Asher et al., 2016) and Molweni (Li et al., 2020). Early approaches to discourse parsing used varied decoding strategies, such as Maximum Spanning Tree (Muller et al., 2012; Afantenos et al., 2012; Li et al., 2014) or Integer Linear Programming (Perret et al., 2016). Researchers soon applied neural models such as Gated Recurrent Units (Shi and Huang, 2019) and Graph Neural Networks (Wang et al., 2021b) to build contextualized embeddings, compared to hand-crafted features from the previous work. More recent works attempted to enhance the parsing task by utilizing Pre-trained Language

Models (PLMs) as backbone (Liu and Chen, 2021; Chi and Rudnicky, 2022), injecting external information such as speaker interactions (Yu et al., 2022; Li et al., 2023b), or joint learning with auxiliary tasks (Yang et al., 2021; He et al., 2021). Due to the small number of annotated examples, some also investigated semi-supervised approaches such as data programming (Badene et al., 2019), bootstrapping (Nishida and Matsumoto, 2022), and signals from the attention matrices in PLMs (Li et al., 2023a). However, much of this line of work dealt only with structure extraction while ignoring relations.

With LLMs on the scene, Chan et al. (2023) evaluated the performance of GPT-3.5 on discourse parsing using zero-shot and few-shot in-context-learning, but only to find that the model performs abysmally. Recently, Maekawa et al. (2024) employed decoder-only LLMs for Rhetorical Structure Theory (RST) discourse parsing in monologues, where conventional top-down and bottom-up strategies are transformed into prompts. On dialogues, only Wang et al. (2023) have investigated discourse parsing with a fine-tuned T5 model. However, their design of output sequences were overly simplified and we observed poor results with a similar abridged scheme in our experiments. In comparison, we explore the effectiveness of using Seq2Seq LLMs for this task with more sophisticated representations, such as an output closer to natural language.

Structure Prediction with Generative Models Loosely related to our work are papers about other structure prediction tasks which also apply generative modeling. For instance, on coreference resolution, Urbizu et al. (2020) conducted a proof-of-concept study where they literally translated the coreference annotation into a target sequence. Zhang et al. (2023) fine-tuned the T0 model with more sophisticated sequence representations that outperformed traditional coreference models. Bohnet et al. (2023) developed a transition-based Seq2Seq system based on mT5, which works on the same principle as our second approach. Paolini et al. (2021) proposed a unified framework that translates a series of structure tasks into *augmented natural languages* using T5. Their work aimed at creating a general and transferable model to solve many tasks. Generative models have also been used for semantic parsing (Rongali et al., 2020), syntactic parsing (He and Choi, 2023), and constituency parsing (Bai et al., 2023). Although large generative models have been successfully applied to various structure prediction tasks, the DDP task, which requires inter-sentence reasoning in dialogues, remains under-explored.

3 A Formal Description of Discourse Parsing and Seq2Seq Modeling

3.1 Discourse Parsing

Given a document $\mathcal{D} = \{e_0, e_1, ..., e_n\}$ where e_i are clause-like text spans known as Elementary Discourse Units (EDU) and e_0 is a dummy *root* node, the general goal of discourse parsing is to create a graph $\mathcal{G}$ composed of (V, E, ℓ) where V is a set of nodes or EDUs including $\{e_0, e_1, ..., e_n\}$, $E_i \subset V \times V$ a set of edges pointing towards the node e_i with $i \in [1, n]$, and ℓ a function $\ell : (e_k, e_i) \mapsto r$ that maps an EDU pair with a rhetorical relation type $r \in \mathbb{R}$, with $0 \leqslant k < i \leqslant n$.

$$E_i = \{(e_k, e_i), e_i \in V, e_k \in V\} \quad (1)$$

Every E_i contains at least one pair of EDUs *pointing to* the node e_i. Here, we emphasize the uni-direction of edges given that in a dialogue, there are no "backwards" edges such that an EDU e_k by speaker a is rhetorically and anaphorically dependent upon a further EDU e_i of speaker b. This is known as *Turn Constraint* in the Segmented Discourse Representation Theory (SDRT) (Asher and Lascarides, 2003; Afantenos et al., 2015). The combination of all E_i is the set of all potential EDU pairs in document $\mathcal{D}$.

$$E = \cup_{i=1}^{n} E_i, \quad E_{\leqslant i} = \cup_{j=1}^{i} E_j \quad (2)$$

The equation 2 defines what we called *discourse structure prediction* where a "naked" graph can be extracted from $\mathcal{D}$. For *full parsing*, each edge must be assigned a relation with the function ℓ. We can expand the pairs in E to triples in F:

$$F_i = \{(e_k, e_i, r_{ki}), e_i \in V, e_k \in V, r_{ki} \in \mathbb{R}\} \quad (3)$$

$$F = \cup_{i=1}^{n} F_i, \quad F_{\leqslant i} = \cup_{j=1}^{i} F_j \quad (4)$$

In a nutshell, discourse parsing takes a document $\mathcal{D}$ as input and predicts the triples F as output. Assuming we have a training set of N examples, $(\mathcal{D}_i, F_i)_{i=1}^{N}$ consists of N pairs of triples.

3.2 Seq2Seq Modeling

Let $\mathcal{V}$ denote the vocabulary. Given a training pair (x, y) where $x \in \mathcal{V}^{T'}$ is the source sequence of length $T' \in \mathbb{N}$, $y \in \mathcal{V}^{T}$ is the target sequence of length $T \in \mathbb{N}$, a Seq2Seq model computes the conditional probability $p(y|x; \theta)$ autoregressively:

$$p(y|x;\theta) = \prod_{t=1}^{T} p(y_t|y_1, ..., y_{t-1}, x; \theta) \quad (5)$$

Model parameters θ are learned by maximizing the sum of conditional probabilities of all examples in the training set:

$$\theta^* = \arg\max_{\theta} \sum_{X,Y} \log p(Y|X;\theta) \quad (6)$$

3.3 Discourse Parsing as Seq2Seq Generation

To conduct discourse parsing with a Seq2Seq model, we translate $(\mathcal{D}, F)$ into a pair of sequences (x, y). The transformation from $\mathcal{D}$ to x is straightforward since $\mathcal{D}$ contains already a sequence of raw text. Our goal is to find a way to express F as a sequence $y \in \mathcal{V}^T$, which is also known as the **"linearization"** process for structured objects. A minimal approach is to literally predict the triples (e_k, e_i, r_{ik}) in F as a sequence of strings. However, such a succinct format performs unsatisfactorily with limited training examples (see analysis in Section 6). We design several representation formats to explore a better solution for structure learning.

Another crucial issue is how to calculate the **conditional probability** $p(y|x)$. We can either feed x all at once and predict y in an end-to-end style or employ a transition system (Nivre, 2008), where the Seq2Seq model takes a single EDU as input and predicts an action corresponding to a set of discourse links involving that EDU as its output. In practice, we implement two Seq2Seq systems: a full text-in text-out system (Section 4) and an improved transition-based system (Section 5).

4 Seq2Seq Modeling for DDP

4.1 Methodology

End-to-End System A Seq2Seq-DDP system takes as input a document with raw text sequences and generates *structure-and-relation-labeled* output for each discourse unit autoregressively. Different from a classic pipeline approach where structure and relation are predicted subsequently (Afantenos et al., 2015; Shi and Huang, 2019; Liu and Chen, 2021; Li et al., 2024), our method jointly predicts link attachment (e_k, e_i) and relation $(e_k, e_i) \mapsto r_{ki}$.

Representation Scheme We investigate two output schemes: a natural scheme and an augmented scheme. For **natural scheme**, we hypothesize that the closer the output is to natural language, the more advantage the Seq2Seq model can take from its pre-training. In other structure prediction tasks such as syntactic dependency parsing (He and Choi, 2023), natural language in the outputs has demonstrated its effectiveness. We use the following as a running example (*pilot01, STAC corpus*):

> $\mathcal{D}$: [e_0] Dave: has anyone got a sheep, [e_1] Dave: I can trade wheat or clay. [e_2] Tomm: Surprisingly I am bereft of sheep.
> F: {(e_0, e_1, Elaboration), (e_0, e_2, QA-pair)}

We describe the triples in F with the template "e_i is r_{ki} of e_k": e_i and e_k are EDU markers; r_{ki} is a relation. In the input, we also append these markers as prefixes for each speech turn. The output joins all sequences with a semicolon. It reads:

> y_{nat}: [e_0] is *root*; [e_1] is Elaboration of [e_0]; [e_2] is Question-Answer-pair of [e_0].

In cases where one node has multiple incoming edges, the template extends its tail to "e_i is r_{ki} of e_k r_{mi} of e_m r_{ni} of e_n", where e_m and e_n (resp. r_{mi} and r_{ni}) are other linked nodes (resp. relations) to e_i. The advantage of this format is that each EDU uses exactly one sentence for structure description so that the length T of prediction y is fixed ($T = T'$).

Inspired by the pioneering work on TANL (Paolini et al., 2021), we design an **augmented scheme** y_{aug} that replicates the input sentences and augments them with link and relation information:

> y_{aug}: [Dave: has anyone got a sheep, | e_0 | *root* = e_0] [Dave: I can trade wheat or clay | e_1 | Elaboration = e_0] [Tomm: Surprisingly I am bereft of sheep. | e_2 | QA-pair = e_0]

Specifically, each EDU is enclosed by the special tokens [ ]. The pipe token | separates raw text, the EDU marker, and a list of relations in the format "$r_{ki} = e_k$". The EDU marker e_i is not prepend in the input. The model needs to use EDU markers to represent utterances and apply them on structure prediction. In other structure prediction tasks such as semantic role labeling (Paolini et al., 2021) and coreference resolution (Zhang et al., 2023; Bohnet et al., 2023), such a representation gives SOTA performance with Seq2Seq models.

Decoding Structured Output Once the model generates an output (y_{nat} or y_{aug}), we decode the sentences to obtain F by following:

- Step1. Split the sequences with semicolons (resp. enclosed brackets) and remove all spe-

	STAC		Molweni		STAC		Molweni	
Scheme	Link	Full	Link	Full	Hallu	Miss	Hallu	Miss
Natural	65.6 ± 0.3	46.9 ± 1.8	81.4 ± 0.4	57.8 ± 0.1	3.1%	1.7%	0.4%	0
Augmented	66.7 ± 0.7	52.0 ± 0.1	82.4 ± 0.4	59.1 ± 1.0	0	0.2%	0	0

Table 1: Seq2Seq-DDP results on STAC and Molweni test sets (left) and error statistics (right). Scores are averaged micro-F_1. "hallu" and "miss": hallucinated and missed EDUs.

cial tokens (*is*, *of*, |, =) to extract triples in y_{nat} and quadruples in y_{aug}.

- Step2. Match the generated $\hat{e}_i$ with the source e_i using heuristics. For y_{nat}, we match EDU markers; for y_{aug}, we match the input sentence and the cleaned output sentence at the token level using the Jaro distance (Jaro, 1989). We use 10 examples from the validation set in STAC and find that using the similarity value > 0.96 can best cover the difference—most of times caused by more spacing between tokens—in generated and gold output. Once the $\hat{e}_i$ and e_i is matched, we obtain the triples in $(e_k, \hat{e}_i, r_{ki})$ which is the predicted structure for EDU e_i.

- Step3. Sanity check for *hallucinated* or *forgotten* EDUs in $\hat{y}$. The output sequence is designed in a way that its length matches the length of the input, so it is easy to spot erroneous generation. We introduce default rules for failure cases: remove the hallucination and add an adjacent attachment with a majority relation (i.e., *Question-answer-pair*) to the missed EDUs[2].

We do not apply constrained decoding (Hokamp and Liu, 2017) as the output is well-aligned with the designed scheme and does not require extra vocabulary masking during generation.

4.2 Experimental Setup

We test our Seq2Seq-DDP system on two most commonly utilized datasets for dialogue discourse parsing: STAC (Asher et al., 2016) is composed of online multi-party conversations during the game *Settlers of Catan*. It contains 1,161 documents with in average 11 speech turns. We follow the subset split in Shi and Huang (2019) and set the maximum document length to 37, resulting in 911, 97, and 109 documents for training, validation, and testing, respectively. Molweni (Li et al., 2020) is a dataset derived from Ubuntu Chat Corpus (Lowe et al., 2015). It contains 10,000 documents with in average 8 utterances. We follow its original separation: 9,000 training, 500 validation, and 500 testing. Both corpora are annotated under the SDRT (Asher and Lascarides, 2003) and have the same relations ($|\mathbb{R}| = 16$). We employ the traditional evaluation metrics, namely, the micro-averaged F_1 scores for link attachment (E) and full structure (F). All our experiments are conducted on T0 model (Sanh et al., 2022) with the 3B checkpoint, without any modification to the architecture. Most hyper-parameters in fine-tuning follow the suggestions in Raffel et al. (2020) (details in Appendix A).

4.3 Results and Analysis

The left part in Table 1 shows the parsing results on STAC and Molweni. Despite the simplicity of the Seq2Seq modeling, the fine-tuned T0 model can well perform dialogue discourse parsing, reaching 66–80 F_1 on the *naked* structure and 47–60 F_1 on the full structure. The outputs are well-aligned with the desired formats and only in rare cases do we observe erroneous generation (see below). Both *natural* and *augmented* formats produce satisfactory results on Molweni (link $F_1 > 81$, full $F_1 > 57$), whereas on STAC, we observe a more pronounced performance difference. The *natural* scheme is a succinct format that utilizes EDU markers in target sequences. This abridgment may cause ambiguity. In fact, the utterances in STAC are short (4.4 tokens/sentence) and similar texts can occur (e.g., the same answer from different speakers towards the same question). In comparison, *augmented* scheme replicates all tokens including speaker markers in the target sequence, helping to reduce ambiguity. Aligned with our observation, Paolini et al. (2021) also reported performance drops when using an abridged format for the entity and relation extraction task.

On the other hand, we observe a few problems originating from the Seq2Seq-DDP design, such as hallucinated or missed EDUs during generation,

[2]In reality, failure cases are few with a $F_1 < \pm 1\%$.

as shown on the right part in Table 1. Since no explicit constraints are placed on the model's output, there is potential for the model to produce invalid EDUs. However, this does not happen often: *natural* scheme generates 3% hallucinated and 1.7% missed EDUs on STAC (resp. 0.4% hallucinated and 0 missed on Molweni); while *augmented* scheme bypasses this issue completely. These erroneous outputs happen typically in longer documents when the number of speech turns exceeds thirty. In practice, we apply refinement rules in post-processing (included in Appendix B) to effectively eliminate this kind of generation.

5 Improve Seq2Seq-DDP Model with Transition-based Algorithm

An inherent drawback of the basic Seq2Seq-DDP system is the weak supervision in long sequences. The longer the document, the harder it is for the model to retrace previous predictions, as evidenced by the hallucinated or forgotten EDUs. Additionally, the act of consecutive output requires extra attention to some properties such as *counting*, which LLMs struggle with (Kojima et al., 2022). To provide more guidance during the generation and bypass the counting issue, we improve the Seq2Seq model with transition-based algorithms. The new Seq2Seq-DDP+Transition system takes a single EDU at each step and predicts an action corresponding to a set of links involving that EDU.

5.1 Methodology

Transition-based System The system we considered is closely related to the deterministic dependency parsing algorithm (Nivre, 2003, 2008). It starts with the dummy *root* e_0 on the stack, all the EDUs in the buffer, and an empty set F. The parse ends once the buffer is empty and F contains triples of all EDUs (Equation 3). The transitions are composed of two actions: *link* action creates a right-arc from one EDU in the stack to the first EDU (i.e., target) in the buffer; *assign* action labels the arc. The target EDU in the buffer is then moved to the stack and a new round of transition will be conducted on the next EDU in the buffer.

States. A state c_i keeps track of which EDU is being processed through the index i, the established pairs $E_{<i}$, and associated relations $F_{<i}$ up to i. We define the following states:

- C is the set of all possible states.
- $c_s = (e_0, \epsilon, \epsilon)$ is the initial state, where two ϵ are the empty sets E and F.
- $C_t = \{c \in C | c = (e_n, E, F)\}$ is the set of the final states.

Actions. Given an intermediate state $c_i = (e_i, E_{<i}, F_{<i})$, we implement a_i which contains a link action $\mathcal{L}(\cdot)$ and an assign action $\mathcal{A}(\cdot)$:

$$\mathcal{L}(e_i, F_{<i}) = \{e_k \rightarrow e_i, 0 \leqslant k < i\} \quad (7)$$

$$\mathcal{A}(e_i, E_i, F_{<i}) = \{(e_k \rightarrow e_i) \mapsto r_{ki}, r \in \mathbb{R}\} \quad (8)$$

The transition function ϕ gives an updated state c_i accordingly:

$$\begin{aligned} &\phi(c_i, (e_k \rightarrow e_i), (e_k \rightarrow e_i) \mapsto r_{ki}) \\ =&(e_i, E_{<i} \oplus (e_k \rightarrow e_i), F_{<i} \oplus r_{ki}) \\ =&(e_i, E_i, F_i) \end{aligned} \quad (9)$$

Our transition system is a quadruple $S = (C, c_s, T, C_t)$ where C, c_s, and C_t are the states defined previously. T is the set of transitions, each of which is a function $\phi : C \rightarrow C$. The parsing path K is a sequence composed of actions and states: $K = \{c_s, a_0, c_1, a_1, ..., c_i, a_i, ..., c_n\}$ where for $i \in [1, n], c_{i+1} = \phi(c_i, a_i)$, and where $a_i = \mathcal{L}_i \cup \mathcal{A}_i, c_n = C_t$.

Representation Scheme Our goal is to encode the parsing path K into input and output strings. Specifically, each state-action pair (c_i, a_i) is mapped to an input-output pair (x_i, y_i). Similar to Seq2Seq-DDP, we design output strings close to natural language. We illustrate two input-output pairs in the **natural scheme**, where the predicted action (underlined) is appended to the next state:

> x_1: [e_0] [Dave: has anyone got a sheep,] is root; [e_1] [Dave: I can trade wheat or clay.] is
> y_{nat_1}: Elaboration of [e_0]
>
> x_2: [e_0] [Dave: has anyone got a sheep,] is root; [e_1] [Dave: I can trade wheat or clay.] is Elaboration of [e_0]; [e_2] [Tomm: Surprisingly I am bereft of sheep.] is
> y_{nat_2}: QA-pair of [e_0].

We also implement a new format called **focused scheme** that utilizes special tokens $**$ to emphasize the target EDU (e_i) and a pipe token | to separate the text with prediction, as depicted in Figure 2.

Decoding and Sliding Window Strategy Compared to the previous system, decoding the structured output from a transition-based model is easier: the generation is incremental with no mismatched or hallucinated EDUs. At each stage, we split $\hat{y}$ on token *of* to obtain e_k and r_{ki}.

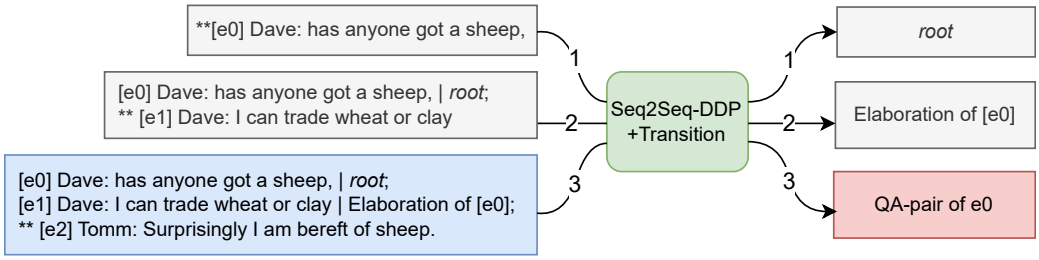

Figure 2: Seq2Seq-DDP+Transition system with *focused* scheme. It takes as input the previous state, the predicted action, and the next EDU; as output, actions for the current state. In blue: current input (c_i); in red: current output (a_i); in grey: parsed input ($C_{<i}$).

The input grows longer as we continue adding the predicted structures. To comply with the maximum input length of pretrained models, we employ a sliding window strategy that reserves the closest EDUs for the next stage of prediction. Naturally, the closest EDUs are most relevant to the target EDU, so we frame a window with a set maximum length and slide it to the right at each stage. We set the window length to 18, as this is the longest link attachment in the validation set. The model is required to focus only on the target EDU e_i and its nearest preceding neighbors in the context c_i[3].

5.2 Experiments and Analysis

We test our new system by fine-tuning T0-3B on STAC and Molweni datasets, results are shown in the first two rows in Table 2. Clearly, the transition-based system outperforms its Seq2Seq-DDP counterpart on all metrics: 5–8 and 1–3 points improvements on STAC and Molweni, respectively.

In the last four rows, we compare with the SOTA models (Shi and Huang, 2019; Liu and Chen, 2021; Chi and Rudnicky, 2022; Li et al., 2023c). Most of which use pre-trained language models such as RoBERTa to provide contextualized representations and task-specific techniques for decoding. Tellingly, our approach obtains new SOTA results on Molweni, surpassing the latest model proposed by Li et al. (2023c). We also achieve comparable results on STAC. Moreover, our approach is not limited to *tree-style* structures. Discourse-aware Seq2Seq models are capable of producing graphs (see Section 6). Although SOTA models use relatively small language models (110M - 340M parameters), it is important to point out that full comparability is challenging due to the numerous ways our approaches differ. First, the complexity of the parsing systems: SOTA models are built upon heavily engineered architecture and require specific decoding strategies such as the Maximum Spanning Tree (MST). Our approach, on the other hand, directly leverages the standard encoder-decoder models and does not require any architecture modification. Second, scaling up encoder-only models does not always result in improvements in downstream applications. These models are also more difficult to deploy. Megatron-BERT (Shoeybi et al., 2019) with 1.3B and 3.9B parameters, for instance, are not publicly available. For generative models consisting of decoder networks, scaling tends instead to be closely associated with improved performance on many tasks (Ganguli et al., 2022).

Compared to Seq2Seq-DDP, the improved system does not suffer from EDU mismatch in the source and generation. However, the model sometimes predicts repetitive structures, such as "Acknowledgment of $[e_2]$ Acknowledgment of $[e_2]$". In reality, failure cases are few: only 13 cases (1%) in all $1.2k$ triples in the development set. This occurs typically when the oracle output contains multiple incoming edges and the model tries to predict a graph structure.

6 Further Investigation

6.1 Masked Labels and Abridged Output

We investigate the influence of label semantics. The semantics of rhetorical relation types can be different in a pre-trained model. To prevent the model from understanding the relation through label semantics, we replace these words with special tokens, such as "rel1" and "rel2", to the model vocabulary. This format is called y_{lmask}:

y_{nat}: $[e_0]$ is *root* ; $[e_1]$ is Elaboration of $[e_0]$; $[e_2]$ is QA-pair of $[e_0]$. *(baseline)*
y_{lmask}: $[e_0]$ is *root* ; $[e_1]$ is rel4 of $[e_0]$; $[e_2]$ is rel0 of $[e_0]$. *(label masked)*

[3]In the oracle structures in test set, the longest edge distance is 13, so this approach does not affect any distant edges.

	System	STAC		Molweni	
		Link (Δ)	Full (Δ)	Link (Δ)	Full (Δ)
Natural (ours)	Seq2Seq-DDP+Transition	70.8 ± 0.9 (↑ 5.2)	55.1 ± 1.0 (↑ 8.2)	83.5 ± 0.2 (↑ 2.1)	60.3 ± 0.1 (↑ 2.5)
Focused (ours)	Seq2Seq-DDP+Transition	72.3 ± 0.6 (↑ 5.5)	56.6 ± 0.6 (↑ 4.6)	83.4 ± 0.6 (↑ 1.0)	60.0 ± 0.5 (↑ 0.9)
Shi and Huang (2019)	GRU+Pointer*	72.9 ± 0.4	54.2 ± 0.5	77.9 ± 0.4	54.1 ± 0.6
Liu and Chen (2021)	RoBERTa+Pointer	72.9 ± 1.5	57.0 ± 1.0	79.0 ± 0.4	55.4 ± 1.8
Chi and Rudnicky (2022)	RoBERTa+CLE†	73.0 ± 0.5	58.1 ± 0.7	81.0 ± 0.7	58.6 ± 0.6
Li et al. (2023c)	BERT+Biaffine+Pointer	73.0	58.5	83.2	59.8

Table 2: Parsing results with our Seq2Seq-DDP+Transition models (top) and replicated SOTA models (bottom) on STAC and Molweni test sets. Scores are averaged micro-F_1. Teal ↑ shows performance gains compared to Seq2Seq-DDP systems. Pointer*: pointer network (Vinyals et al., 2015). CLE†: Chu-Liu-Edmonds algorithm (Chu, 1965; Edmonds et al., 1967).

Additionally, to analyze the impact of sequence representations, we design abridged formats (y_{abr}) for *natural* and *augmented* schema:

y_{nat}: [e_0] is *root* ; [e_1] is Elaboration of [e_0] ; [e_2] is QA-pair of [e_0]. *(baseline)*
y_{abr}: [e_0] *root*; [e_1] [e_0] rel4; [e_2] [e_0] rel0. *(abridged)*

y_{aug}: [Dave: has anyone got a sheep, | e_0 | *root* = e_0] [Dave: I can trade wheat or clay | e_1 | Elaboration = e_0] [Tomm: Surprisingly I am bereft of sheep. | e_2 | QA-pair = e_0] *(baseline)*
y_{abr}: e_0 | *root* = e_0 ; e_1 | Elaboration = e_0 ; e_2 | QA-pair = e_0. *(abridged)*

For the abridged version of *natural* representation, we transform the output into a triple (x, y, r) where x and y are respectively the dependent and head of an EDU pair; r is the masked relation type. It reads: EDU x is linked to EDU y with relation r. This is the expected output F from document $\mathcal{D}$ (Equation 3), but such an extremely short linearization creates the most challenging representation: the model not only needs to learn the semantics of masked labels but also the implicit output pattern. For the abridged version of *augmented* representation, we do not repeat the input utterance and only keep EDU markers. The pipe (|) tag still denotes the start of the area of interest. Without the original text sequence, the abridged scheme requires extra reasoning to map the text with EDU markers.

We present the results of masked labels and abridged output in Table 3. On STAC, masking out the labels substantially hurt the performance with -2.5 points in link prediction and -9.6 in full. This demonstrates that label semantics are useful, especially for datasets containing smaller training examples. In terms of abridged output, both *natural abridged* and *augmented abridged* formulations underperform the baselines significantly (-12 and -9.7 points on full prediction). Interestingly, we do not observe a similar performance drop on Molweni. Label-masked models obtain similar results as the *natural* baseline. The differences in link and full gains are not significant: $p > 0.7$, $p > 0.4$. The most challenging abridged formulation also continues to perform well on Molweni. We think the amount of supervision is key. Molweni contains $9,000$ documents in the training set whereas STAC only ≈ 900. In terms of utterance length and token number, STAC is also very limited (see Table 5). These results are informative, indicating that a more "natural language"–like output generally brings more accurate predictions, especially when the amount of training data is low. On the other hand, sufficient supervision enables us to use the simpler paradigm of a text-to-text model successfully.

6.2 Pretrained LLMs and Model Sizes

We compare three LLMs in the T5 family: T5 (Raffel et al., 2020), Flan-T5 (Chung et al., 2022), and T0 (Sanh et al., 2022). In Table 4, we find that the model performance improves as the model size increases, which in line with the observations in Zhang et al. (2023). In terms of different models in the T5 family, there is a notable difference between models with and without instruction finetuning such as FLAN (Wei et al., 2022). For models of the same size, the performance of the Flan-T5 and T0 is comparable (link 68.5 vs. 69.2; full 50.4 vs. 50.2), and both greatly exceed the performance of the original T5 model ($+8$ points in link attachment and $+10$ points in full prediction). Even the much smaller Flan-T5-base model (250M) outperforms T5-3B on link prediction by 2 points. This is not surprising: Chung et al. (2022) demonstrate that on some challenging BIG-Bench tasks (Srivastava et al., 2023), Flan-T5-11B outperforms the same size T5 by double-digit performances. This proves that instruction tuning can significantly enhance

	STAC		Molweni	
Sequence representation	Link (F_1)	Full (F_1)	Link (F_1)	Full (F_1)
Natural baseline	69.2 ± 0.5	50.2 ± 0.7	83.2 ± 1.4	58.6 ± 0.8
Label masked	$\downarrow -2.5 \pm 0.9$	$\downarrow -9.6 \pm 0.4$	$\uparrow +0.3 \pm 0.4$	$\uparrow +0.6 \pm 0.5$
Label masked + abridged	$\downarrow -2.7 \pm 0.2$	$\downarrow -12.4 \pm 3.0$	$\uparrow +1.3 \pm 1.0$	$\uparrow +0.6 \pm 0.2$
Augmented baseline	70.0 ± 0.8	54.2 ± 0.4	84.5 ± 0.4	59.0 ± 1.0
Abridged	$\downarrow -2.6 \pm 0.9$	$\downarrow -9.7 \pm 0.4$	$\sim \pm 0.9$	$\uparrow +0.7 \pm 1.1$

Table 3: Sequence representation study on STAC and Molweni development sets. Red ↓, teal ↑, and ∼ symbols refer to resp. lower, higher, and same scores compared to the baselines.

Pre-trained model	#Params	Link (F_1)	Full (F_1)
T5-large	738M	59.3 ± 0.6	36.4 ± 0.6
T5-3B	3B	60.7 ± 1.3	40.5 ± 0.9
Flan-T5-base	250M	63.0 ± 0.5	36.7 ± 0.1
Flan-T5-large	780M	67.2 ± 1.4	46.6 ± 1.8
Flan-T5-xl	3B	$\underline{68.5} \pm 0.5$	$\mathbf{50.4} \pm 0.1$
T0-3B	3B	$\mathbf{69.2} \pm 0.5$	$\underline{50.2} \pm 0.7$

Table 4: Study of different models in the T5 family on STAC development set (*natural* scheme). The best and second-best scores are **bolded** and underlined.

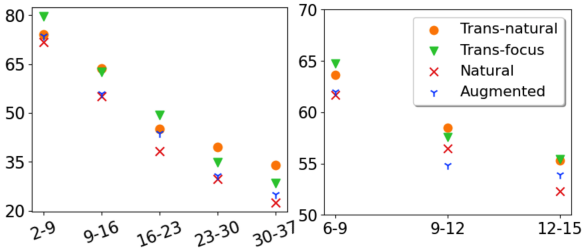

Figure 3: STAC (left) and Molweni (right) Full parsing performance under different Seq2Seq models and document lengths. x axis: #EDUs in a document. y axis: F1.

the model's ability to learn complex language tasks, such as dialogue discourse parsing, thereby advancing it towards human-like language reasoning.

6.3 Richer Output Structures

We observe some distinctive features in the predicted structures such as directed acyclic graphs with Seq2Seq models. This is an exciting and big advantage over other SOTA models (Shi and Huang, 2019; Liu and Chen, 2021; Wang et al., 2021a; Chi and Rudnicky, 2022; Li et al., 2023a) that can only generate trees using MST algorithms in decoding (Eisner, 1996; Chu, 1965; Edmonds et al., 1967). Among all the proposed schemes, the *focused* scheme in Seq2Seq-DDP+Transition system achieves the highest performance in capturing multiple incoming edges, with a precision rate of 13% for graph structures. Other schemes such as *natural* and *augmented* also correctly predict around 10% graph structures. This is non-trivial: these structures are few and difficult to learn ($\approx 5\%$ of nodes, $< 7\%$ of links in STAC; none in Molweni) and demonstrate interesting and unique structures in dialogues.

6.4 Different Document Lengths

Since long documents can pose challenges for Seq2Seq models, we analyze the parsing performance under different document lengths, as shown in Figure 3. On STAC, we split the length range into five even buckets between the shortest (2 EDUs) and longest (37 EDUs) document, resulting in 60, 25, 16, 4, and 4 data points per bucket. On Molweni, we split the documents into three buckets with 276, 154, and 70 data points in each group. Both the Seq2Seq-DDP and Seq2Seq-DDP+Transition systems exhibit a decline in performance with longer documents. However, our transition-based models ("Trans-*") show a superior ability to handle long documents compared to their counterparts, as validated across both datasets.

7 Conclusion

We investigate an effective transformation approach for the DDP task by leveraging Seq2Seq LLMs. We adopt the pretrained encoder-decoder model T0 and fine-tune it to produce structured sequences. Without using any specific parsing module or modifying LLM architecture, our Seq2Seq-DDP system performs reasonably well on STAC and Molweni datasets. Excitingly, our Seq2Seq-DDP+Transition system yields comparable results with task-specific SOTA models, with richer discourse structures. Building on this work, we intend to explore various generative model architectures and sequence representations, and eventually extend our method to other discourse parsing tasks.

Limitations

Longer documents tend to be more difficult to parse due to the growing number of possible discourse parse trees and the inherent drawbacks such as *counting* in LLMs. Our Transition-based systems mitigate this issue to some extent by using a sliding window strategy that focuses only on the closest EDUs.

In terms of decoding speed and performance, end2end systems demonstrate lower F_1 score but faster inference compared to transition-based systems. On the development set of STAC, the inference time for the end2end system is 2.5 seconds per document, whereas the transition-based system takes around 1.8 seconds per sequence, summing up to around 20 seconds for a complete document prediction.

Acknowledgments

We thank the anonymous reviewers for their insightful comments and suggestions. We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). Nous remercions le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) de son soutien. The computing resources are provided by the Digital Research Alliance of Canada (alliancecan.ca).

References

Stergos Afantenos, Nicholas Asher, Farah Benamara, Anaıs 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 the 16th Workshop on the Semantics and Pragmatics of Dialogue (Seinedial), Paris*.

Stergos Afantenos, Eric Kow, Nicholas Asher, and Jérémy Perret. 2015. Discourse parsing for multiparty chat dialogues. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 928–937, Lisbon, Portugal. Association for Computational Linguistics.

Nicholas Asher, Julie Hunter, Mathieu Morey, Benamara Farah, and Stergos Afantenos. 2016. Discourse structure and dialogue acts in multiparty dialogue: the STAC corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2721–2727, Portorož, Slovenia. 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. 2019. Data programming for learning discourse structure. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 640–645, Florence, Italy. Association for Computational Linguistics.

Xuefeng Bai, Jialong Wu, Yulong Chen, Zhongqing Wang, and Yue Zhang. 2023. Constituency parsing using llms. *arXiv preprint arXiv:2310.19462*.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali. Association for Computational Linguistics.

Bernd Bohnet, Chris Alberti, and Michael Collins. 2023. Coreference resolution through a seq2seq transition-based system. *Transactions of the Association for Computational Linguistics*, 11:212–226.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

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. *arXiv preprint arXiv:2304.14827*.

Ta-Chung Chi and Alexander Rudnicky. 2022. Structured dialogue discourse parsing. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 325–335.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language

modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Yoeng-Jin Chu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jack Edmonds et al. 1967. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240.

Jason M. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2021. A survey on dialogue summarization: Recent advances and new frontiers. *arXiv preprint arXiv:2107.03175*.

Deep Ganguli, Danny Hernandez, Liane Lovitt, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova Dassarma, Dawn Drain, Nelson Elhage, et al. 2022. Predictability and surprise in large generative models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1747–1764.

Ivan Habernal, Omnia Zayed, and Iryna Gurevych. 2016. C4Corpus: Multilingual web-size corpus with free license. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 914–922, Portorož, Slovenia. European Language Resources Association (ELRA).

Han He and Jinho D Choi. 2023. Unleashing the true potential of sequence-to-sequence models for sequence tagging and structure parsing. *Transactions of the Association for Computational Linguistics*, 11:582–599.

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*, pages 551–561, Shanghai, China. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Matthew A Jaro. 1989. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Chuyuan Li, Chloé Braud, Maxime Amblard, and Giuseppe Carenini. 2024. 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, St. Julians, Malta. Association for Computational Linguistics.

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

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

Jiaqi Li, Ming Liu, Yuxin Wang, Daxing Zhang, and Bing Qin. 2023b. A speaker-aware multiparty dialogue discourse parser with heterogeneous graph neural network. *Cognitive Systems Research*, 79:15–23.

Jiwei Li, Rumeng Li, and Eduard Hovy. 2014. Recursive deep models for discourse parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2061–2069.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019.

Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhengyuan Liu and Nancy Chen. 2021. Improving multi-party dialogue discourse parsing via domain integration. In *Proceedings of the 2nd Workshop on Computational Approaches to Discourse*, pages 122–127, Punta Cana, Dominican Republic and Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

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 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294, Prague, Czech Republic. Association for Computational Linguistics.

Xinbei Ma, Zhuosheng Zhang, and Hai Zhao. 2023. Enhanced speaker-aware multi-party multi-turn dialogue comprehension. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Aru Maekawa, Tsutomu Hirao, Hidetaka Kamigaito, and Manabu Okumura. 2024. Can we obtain significant success in RST discourse parsing by using large language models? In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2803–2815, St. Julian's, Malta. Association for Computational Linguistics.

Philippe Muller, Stergos Afantenos, Pascal Denis, and Nicholas Asher. 2012. Constrained decoding for text-level discourse parsing. In *Proceedings of COLING 2012*, pages 1883–1900, Mumbai, India. The COLING 2012 Organizing Committee.

Noriki Nishida and Yuji Matsumoto. 2022. Out-of-domain discourse dependency parsing via bootstrapping: An empirical analysis on its effectiveness and limitation. *Transactions of the Association for Computational Linguistics*, 10:127–144.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 149–160, Nancy, France.

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Subendhu Rongali, Luca Soldaini, Emilio Monti, and Wael Hamza. 2020. Don't parse, generate! A sequence to sequence architecture for task-oriented semantic parsing. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2962–2968. ACM / IW3C2.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Zhouxing Shi and Minlie Huang. 2019. A deep sequential model for discourse parsing on multi-party dialogues. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7007–7014.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adri Garriga-Alonso, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Gorka Urbizu, Ander Soraluze, and Olatz Arregi. 2020. Sequence to sequence coreference resolution. In *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 39–46, Barcelona, Spain (online). Association for Computational Linguistics.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems*, 28.

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

Ante Wang, Linfeng Song, Lifeng Jin, Junfeng Yao, Haitao Mi, Chen Lin, Jinsong Su, and Dong Yu. 2023. D 2 psg: Multi-party dialogue discourse parsing as sequence generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Jinfeng Wang, Longyin Zhang, and Fang Kong. 2021b. Multi-level cohesion information modeling for better written and dialogue discourse parsing. In *Natural Language Processing and Chinese Computing - 10th CCF International Conference, NLPCC 2021, Qingdao, China, October 13-17, 2021, Proceedings, Part I*, volume 13028 of *Lecture Notes in Computer Science*, pages 40–52. Springer.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jingxuan Yang, Kerui Xu, Jun Xu, Si Li, Sheng Gao, Jun Guo, Nianwen Xue, and Ji-Rong Wen. 2021. A joint model for dropped pronoun recovery and conversational discourse parsing in Chinese conversational speech. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1752–1763, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Nan Yu, Guohong Fu, and Min Zhang. 2022. Speaker-aware discourse parsing on multi-party dialogues. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5372–5382, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Wenzheng Zhang, Sam Wiseman, and Karl Stratos. 2023. Seq2seq is all you need for coreference resolution. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11493–11504, Singapore. Association for Computational Linguistics.

A Experimental Setup

The data statistics are given in Table 5. All our experiments are conducted on T0 model (Sanh et al., 2022) with the 3B checkpoint: `https://huggingface.co/bigscience/T0_3B`. The hyper-parameters for fine-tuning are kept as simple as possible. We do not apply parameter efficient fine-tuning techniques nor use lower precision during training. We apply a constant learning rate ($5e$–5) using the AdamW optimizer (Loshchilov and Hutter, 2018). The mini-batch sizes are set to 4 for both *natural* and *augmented* schemes. The maximum input and output lengths are set to 512 and 1024. To fit in the positional embeddings of T0, we discard 36 and 6 documents in the STAC train and development sets, respectively. The actual training and development sets thus contain 911 and 97 documents, respectively. The test set is not affected. No document is discard for Molweni. On Seq2Seq-DDP system, we train for a maximum of 20 epochs on STAC (resp. 10 epochs on Molweni) for 3B models, which takes about 5 hours (resp. 13 hours) on 1 A100 80G GPU. On Seq2Seq-DDP+Transition system, we train for a maximum of 10 epochs on STAC (resp. 5 epochs on Molweni), which takes around 12 hours (resp. 60 hours).

B Seq2Seq-DDP System Examples of Erroneous Generation

Table 6 presents a few concrete examples of the error generations using Seq2Seq-DDP system. Specifically, we find that in *natural* scheme, 38 EDUs from 19 documents are hallucinated; 22 EDUs are not predicted (missed) in 11 documents, all of which have an EDU index greater than 18. In the *augmented* scheme, there are no hallucinated EDUs; 3 EDUs are not predicted. The typical error in this format comes from the wrong counting of large EDU index, as shown in orange in the Table.

	Train			Development			Test		
Dataset	#Doc	#Sent	#Token	#Doc	#Sent	#Token	#Doc	#Sent	#Token
STAC	911	$10k$	$47k$	97	$1k$	$5k$	109	$1k$	$5k$
Molweni	9000	$79k$	$945k$	500	$4k$	$52k$	500	$4k$	$52k$

Table 5: Data statistics for STAC and Molweni corpora. The numbers of documents, utterances, and tokens in training, development, and test sets are listed.

Error	x	y	$\hat{y}$
Hallu	x_{nat}: ...[e_{13}] Gaeilgeoir: I'll try again [e_{14}] nareik15: cool :) [e_{15}] Gaeilgeoir: I'm definitely giving wheat [e_{16}] Gaeilgeoir: I have no sheep :)	y_{nat}: ...[e_{14}] is Acknowledgement of [e_{13}] ; [e_{15}] is Continuation of [e_{13}] ; [e_{16}] is Elaboration of [e_{15}].	$\hat{y}_{nat}$: [e_{14}] is Acknowledgement of [e_{12}] ; [e_{15}] is Result of [e_{14}] ; [e_{16}] is QA-pair of [e_{15}] ; [e_{17}] is Contrast of [e_{16}].
Miss	x_{nat}: [e_0] ztime: morning. [e_1] Shawnus: hey [e_2] Shawnus: good morning ...[e_{28}] ztime: damn [e_{29}] Shawnus: misplaced/ [e_{30}] Shawnus: ? [e_{31}] somdechn: Need to undo are you? [e_{32}] ztime: no. [e_{33}] ztime: you took the spot I was looking at. [e_{34}] ztime: no it's fine. [e_{35}] Shawnus: hahaha [e_{36}] somdechn: Got to be mean here.	y_{nat}: [e_0] is root; [e_1] is Acknowledgement of [e_0] ; [e_2] is Elaboration of [e_1] ; ...[e_{29}] is Clarification_question of [e_{28}] ; [e_{30}] is Correction of [e_{29}] ; [e_{31}] is Clarification_question of [e_{28}] ; [e_{32}] is QA-pair of [e_{29}] ; [e_{33}] is Explanation of [e_{32}] ; [e_{34}] is QA-pair of [e_{31}] ; [e_{35}] is Comment of [e_{32}] ; [e_{36}] is Comment of [e_{32}].	$\hat{y}_{nat}$: [e_0] is root; [e_1] is Acknowledgement of [e_0] ; [e_2] is Continuation of [e_1] ; ...[e_{29}] is Comment of [e_{28}] ; [e_{30}] is Comment of [e_{28}] ; [e_{30}] is Comment of [e_{28}] ; [e_{30}] is Comment of [e_{28}] ; [e_{30}] is Comment of [e_{28}]
Count	x_{aug}: [ztime: morning] [Shawnus: hey] [Shawnus: good morning] ...[ztime: damn] [Shawnus: misplaced/] [Shawnus: ?] [somdechn: Need to undo are you?] [ztime: no..] [ztime: you took the spot I was looking at.] [ztime: no it's fine] [Shawnus: hahaha] [somdechn: Got to be mean here.]	y_{aug}: [ztime: morning | e_1 | root = e_0] [Shawnus: hey | e_1 | Acknowledgement = e_0] [Shawnus: good morning | e_2 | Elaboration = e_1] ...[Shawnus: misplaced/ | e_{29} | Clarification_question = e_{28}] [Shawnus: ? | e_{30} | Correction = e_{29}] [somdechn: Need to undo are you? | e_{31} | Clarification_question = e_{28}] [ztime: no. | e_{32} | QA-pair = e_{29}] [ztime: you took the spot I was looking at. | e_{33} | Explanation = e_{32}] [ztime: no it's fine. | e_{34} | QA-pair = e_{31}] [Shawnus: hahaha | e_{35} | Comment = e_{32}] [somdechn: Got to be mean here. | e_{36} | Comment = e_{32}]	$\hat{y}_{aug}$: [ztime: morning | e_1 | root = e_0] [Shawnus: hey | e_1 | Acknowledgement = e_0] [Shawnus: good morning | e_2 | Continuation = e_1] ...[Shawnus: misplaced/ | e_{25} | QA-pair = e_{24}] [Shawnus:? | e_{25} | Continuation = e_{24}] [somdechn: Need to undo are you? | e_{25} | Clarification_question = e_{24}] [ztime: no. | e_{25} | QA-pair = e_{24}] [ztime: you took the spot I was looking at. | e_{25} | Explanation = e_{24}] [ztime: no it's fine. | e_{25} | Acknowledgement = e_{24}] [Shawnus: hahaha | e_{25} | Comment = e_{24}] [Shawnus: hahaha | e_{27} | Comment = e_{24}] [Shawnus: hahaha | e_{27}

Table 6: Error generation examples in STAC corpus. x, y, $\hat{y}$ refer to resp. source input, target output, and generated output. "Hallu": hallucinated EDU in teal; "Miss": missing EDUs in cyan; "Count": wrong counting of EDU index in orange. False predictions are in red.