D²PSG: Multi-Party Dialogue Discourse Parsing as Sequence Generation

Ante Wang, Linfeng Song, Lifeng Jin, Junfeng Yao, Haitao Mi, Chen Lin, *Member, IEEE*, Jinsong Su, and Dong Yu, *Fellow, IEEE*

***Abstract*—Conversational discourse analysis aims to extract the interactions between dialogue turns, which is crucial for modeling complex multi-party dialogues. As the benchmarks are still limited in size and human annotations are costly, the current standard approaches apply pretrained language models, but they still require randomly initialized classifiers to make predictions. These classifiers usually require massive data to work smoothly with the pretrained encoder, causing severe data hunger issue. We propose two convenient strategies to formulate this task as a sequence generation problem, where classifier decisions are carefully converted into sequence of tokens. We then adopt a pretrained T5 [C. Raffel et al., 2020] model to solve this task so that no parameters are randomly initialized. We also leverage the descriptions of the discourse relations to help model understand their meanings. Experiments on two popular benchmarks show that our approach outperforms previous state-of-the-art models by a large margin, and it is also more robust in zero-shot and few-shot settings.[1]**

***Index Terms*—Multi-party dialogue discourse parsing, pretrained language model, model initialization, sequence generation.**

I. Introduction

Recent years have witnessed a surge of interest in modeling dialogues that usually involve two or more speakers. For multi-party dialogues, the task of dialogue discourse parsing has been proposed to discover the intercorrelation in each pair of dialogue turns.[2] This is crucial because multiple speakers are involved, adding extra complexity to the dialogue flow.

Manuscript received 29 November 2022; revised 29 June 2023; accepted 3 September 2023. Date of publication 15 September 2023; date of current version 20 October 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62276219, in part by the Natural Science Foundation of Fujian Province of China under Grant 2020J06001, and in part by Youth Innovation Fund of Xiamen under Grant 3502Z20206059. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yun-Nung Chen. *(Corresponding author: Jinsong Su.)*

Ante Wang, Junfeng Yao, and Jinsong Su are with the School of Informatics, Xiamen University, Xiamen 361000, China, and also with the Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan (Xiamen University), Ministry of Culture and Tourism, Fujian 361005, China (e-mail: wangante@stu.xmu.edu.cn; yao0010@xmu.edu.cn; jssu@xmu.edu.cn).

Linfeng Song, Lifeng Jin, Haitao Mi, and Dong Yu are with the Tencent, AI Lab, Bellevue, WA 98004 USA (e-mail: freesunshine0316@gmail.com; lifengjin@gmail.com; haitaominlp@gmail.com; dongyu@ieee.org).

Chen Lin is with the School of Informatics, Xiamen University, Xiamen 361000, China (e-mail: chenlin@xmu.edu.cn).

Digital Object Identifier 10.1109/TASLP.2023.3313415

[1]https://github.com/DeepLearnXMU/D2PSG

[2]In dialogue discourse parsing, each turn (utterance) is an elementary discourse unit (EDU).

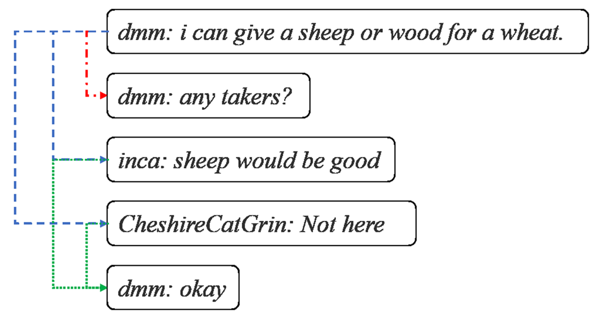

Fig. 1. Multi-party dialogue from the *STAC* dataset [8] with its discourse structure, where the links in slash blue, slash-dotted red and dotted green denote "Question-Answer Pair", "Q-Elab", and "Acknowledgement" respectively. .

Fig. 1 shows a multi-party conversation of three speakers (*dmm, inca, CheshireCatGrin*) and the corresponding discourse structure. We can observe that the discourse structure effectively represents the relations between non-adjacent utterances, such as the "*Question-Answer Pair*" relation between the first turn and the fourth turn in the dialogue. Incorporating conversational discourse information has been proven beneficial for various downstream tasks, such as dialogue response generation [2], summarization [3], [4] and question answering [5], [6], [7].

Most previous efforts [7], [9], [10], [11], [12] formulate the prediction of each discourse relation as two *classification* steps: for each utterance pair (e.g., the first and the fourth one in Fig. 1), they first decide whether this pair forms a discourse relation, before predicting the corresponding relation type. Both types of predictions are conducted by using separate classifiers that take the utterance representations as inputs. To effectively encode the information from dialogue context, most previous work [9], [10], [11], [12], [13] adopts a hierarchical encoder, where each utterance is firstly presented by a recurrent neural network (RNN) or a Transformer [14] encoder, then the encoding outputs are fed into another utterance-level RNN or Transformer to get context-aware representations. Besides, most previous work [10], [11], [12], [13] solves this task in the *offline* manner, where the context of the whole dialogue is required to make classification decisions for the intermediate dialogue turns. This limits the usability of dialogue discourse parsing on important applications like online chatbots.

Whereas the burgeoning of pretrained language models (LMs) across various NLP tasks, previous work [7], [10], [11], [12], [13] has shown that using a pretrained LM as the sentence

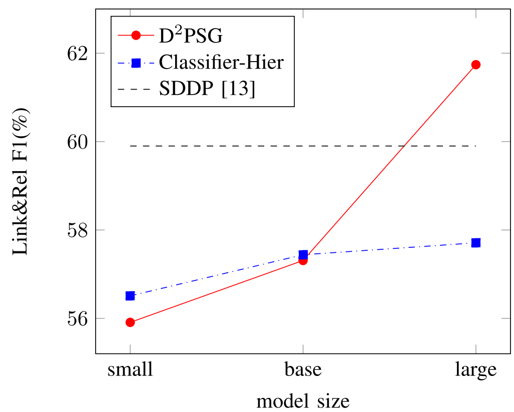

Fig. 2. Performances of a *Classifier-Hier* baseline and our model regarding model size on *Molweni*. They both use a pretrained T5 [1] model as backbone. *SDDP* proposed in the latest work [13] is the previous SOTA model, but it works in the *offline* manner, requiring full dialogue context to make classification decisions.

encoder can be significantly beneficial. However, we find that the performance gain by enlarging the size of the pretrained LM is very marginal. As shown in Fig. 2 (blue slashed line), though T5-large is 10-time larger than T5-small, it only gives an increase of 1.2 F_1 points on the *Molweni* benchmark [5]. The reason is that the utterance-level encoder and the classifiers are still trained from scratch, thus they cannot fully exploit the rich features from the pretrained sentence encoder by being tuned only on limited benchmark data. This causes the data hunger issue.

We propose to formulate this task as a sequence generation problem so that a pretrained encoder-decoder model can be directly applied without the need of adding any randomly initialized classifiers. To this end, we introduce two effective strategies to linearize the classification decisions of dialogue discourse parsing into token sequences. Taking concatenated history utterances as inputs, the first strategy only casts the discourse-classification decisions of the latest turn, while the second strategy casts the decisions of all dialogue turns in natural order. Using Fig. 1 as the example, the token sequences generated by the strategies are "*T4, T3: Acknowledgement*" and "*T1, T0: Q-Elab; T2, T0: Question-answer pair; T3, T0: Question-answer pair; T4, T3: Acknowledgement*", respectively. Comparing with the first strategy, the second one can leverage additional context but with *extra* noise. In addition, we also leverage the description of each relation type as extra inputs to help model better understand the discourse relations.

We then build D²PSG, a pretrained T5 [1] model with constrained decoding to generate legal sequences under our proposed strategies. Different from most previous approaches that work in an offline manner, D²PSG analyzes each ongoing dialogue, making it more broadly applicable than these approaches.

Experiments on two popular benchmarks show that our model (D²PSG) significantly outperforms previous state-of-the-art (SOTA) systems, and its performance can be effectively improved by enlarging model size as shown in Fig. 2. To validate the generalization capability of our model, we conduct cross domain zero-shot transfer evaluation as [11], and we further evaluate on few-shot setting and long-tail cases of this task, which have not been explored in previous work. In-depth analyses show that enlarging model scale has less benefit on previous approaches and even hurt their model performances under extreme settings, while our models are more robust and can always benefit from a larger pretrained model.

II. Problem Definition

Formally, for each EDU (utterance) x_i in a sequence of EDUs $x_1, x_2, \ldots, x_N$ from a dialogue, the goal is to pick a target EDU x_j from all antecedent EDUs ($x_{<i}$) of x_i and to decide their discourse type. Generally, the prediction of each discourse relation (x_j, x_i, r_{ji}) is divided into *link prediction* $P(x_j \rightarrow x_i \mid x_0, x_1, \ldots, x_i)$ and *relation classification* $P(r_{ji} \mid x_j \rightarrow x_i)$.

III. Baselines

In this section, we describe two baseline systems (*Classifier-Hier* and *Classifier-Concat*), which cover the previous efforts on neural conversational discourse parsing.

A. The Hierarchical Encoder Baseline

Using a hierarchical encoder [15], [16], [17] has become popular for representing a dialogue context, including multiple previous efforts [9], [10], [11], [12], [13] on dialogue discourse parsing.

We follow these efforts to build the *Classifier-Hier* baseline. In particular, a Graph Transformer [18], [19] is adopted as the dialogue-level encoder, and it takes the utterance representations produced by an utterance-level encoder. To be consistent with our model, we use a T5 [1] encoder with mean pooling as the utterance-level encoder to calculate the representation vector $\boldsymbol{u}_i^{(0)}$ of each utterance x_i:

$$\boldsymbol{u}_i^{(0)} = \texttt{MeanPool}\left(\texttt{T5}-\texttt{Enc}(x_i)\right) \tag{1}$$

A Graph Transformer of T layers is then used to update the initial utterance representations (e.g., $\boldsymbol{u}_i^{(0)}$) with more global information. Following previous work, each input graph is fully connected with the utterances as its nodes. The label (e.g., $\boldsymbol{\varepsilon}_{ij}$) of each edge contains the speaker and relative position information between the utterances it connects. The Graph Transformer takes a similar structure with a vanilla Transformer [14], but it adopts relation-aware self-attention (instead of vanilla self-attention) defined below:

$$\begin{aligned}
\boldsymbol{u}_i^{(t+1)} &= \sum_{j=1}^{N} \boldsymbol{\alpha}_{ij} \left(\boldsymbol{u}_j^{(t)} \boldsymbol{W}^V + \boldsymbol{\varepsilon}_{ij} \boldsymbol{W}^F\right), \\
\boldsymbol{\alpha}_{ij} &= \frac{\exp\left(\boldsymbol{e}_{ij}\right)}{\sum_{j'=1}^{i} \exp\left(\boldsymbol{e}_{ij'}\right)}, \\
\boldsymbol{e}_{ij} &= \frac{\left(\boldsymbol{u}_i^{(t)} \boldsymbol{W}^Q\right) \left(\boldsymbol{u}_j^{(t)} \boldsymbol{W}^K + \boldsymbol{\varepsilon}_{ij} \boldsymbol{W}^R\right)^{\intercal}}{\sqrt{d_u}},
\end{aligned} \tag{2}$$

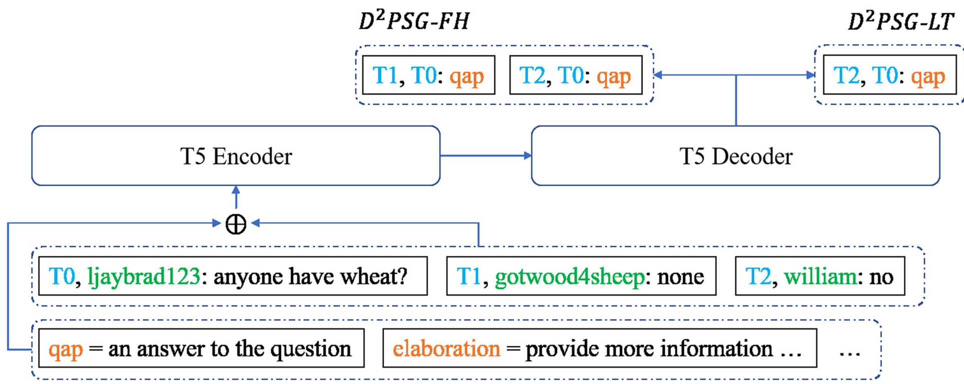

Fig. 3. Model illustration with an example of 3 turns. The turn marker, speaker and relation type are in different colors for better understanding.

where ε_{ij} is the embedding vector for the edge connecting x_i and x_j, $\boldsymbol{u}_i^{(t)}$ is the i-th utterance representation at the t-th layer, and all the $\boldsymbol{W}^x$ are model parameters.

Finally, we define the feature vector $\boldsymbol{F}_{i,j}$ between x_i and x_j as $[\boldsymbol{u}_i^{(0)};\boldsymbol{u}_i^{(T)};\boldsymbol{u}_j^{(0)};\boldsymbol{u}_j^{(T)}]$, which is taken as the input of the linear classifiers for relation link and type classifications. For example, the loss terms for x_i are:

$$\mathcal{L}_i = \mathcal{L}_i^{link} + \mathcal{L}_i^{rel},$$

$$\mathcal{L}_i^{link} = -\sum_{j=1}^{i-1} \log P_{link}\left(x_i^* == x_j \mid \boldsymbol{F}_{i,j}\right),$$

$$\mathcal{L}_i^{rel} = -\log P_{rel}\left(r_{i,j}^* \mid \boldsymbol{F}_{i,j}\right), \quad (3)$$

where x_i^* and $r_{i,j}^*$ denote the gold discourse target and the corresponding relation for x_i, respectively, and $x_i^* == x_j$ is an indicator on whether x_j is the gold discourse target of x_i. Note that if x_i does not depend on any preceding utterance (e.g., being the first utterance), then $x_i^* = x_i$ and $r_{i,j}^* = \texttt{none}$.

B. The Flat Encoder Baseline

Another line of research [7] suggests concatenating dialogue utterances into a long sequence, which is then fed into a pretrained encoder. Following this line of research, we build the *Classifier-Concat* baseline that concatenates all history utterances as inputs to a T5 encoder. It then takes the hidden state of the special token (`[SEP]`) after each utterance as its representation:

$$\boldsymbol{u}_1, \ldots, \boldsymbol{u}_i = \texttt{T5}-\texttt{Enc}\left(x_1[\text{SEP}]\ldots x_i[\text{SEP}]\right) \quad (4)$$

As the next step, we follow [7] to get the feature vector $F_{i,j} = (\boldsymbol{u}_i, \boldsymbol{u}_j, \boldsymbol{u}_i - \boldsymbol{u}_j, \boldsymbol{u}_i \cdot \boldsymbol{u}_j)$, which are taken as the inputs of the final. Similar with *Classifier-Hier* (Section III-A), linear classifiers and the same loss functions ((3)) are adopted for fair comparison.

C. Comparison and Discussion

Comparing with *Classifier-Hier*, *Classifier-Concat* may better capture the global correlations from token-level information mix through the pretrained self-attention-based encoder. However, it consumes more memory than *Classifier-Hier* and may exceed the maximum supported length (typically 512) of its encoder. As a common issue, they both contain randomly initialized parameters. This can cause data hunger, as more training data is required to train a robust module from scratch. Some popular approaches can be adopted to ease this problem such as meta learning [20] and knowledge distillation [21]. In this work, we solve this issue using better initialization with an Encoder-Decoder pretrained model.

IV. Approach

As shown in Fig. 3, our model consumes a dialogue history and directly generates the dependency discourse relations. Particularly, in Section IV-A, we propose turn markers and exploit two prediction strategies to formulate this task as sequence generation. Then, we describe our model structure in Section IV-B and further extend our method with task descriptions in Section IV-C.

A. Classification as Sequence Generation

Different from other typical classification tasks, the major problem here is: *How to express the structural information of typed links connecting pairs of utterances from dialogue context in a sequence?*

In this work, we propose using a special turn marker to resolve this problem. Particularly, we first introduce a turn marker (e.g., T_i) before each (e.g., the i-th) utterance to indicate its position in a dialogue. Then, an input dialogue $x_1, x_2, \ldots, x_N$ with N utterances can be converted into $T_1, x_1, T_2, x_2, \ldots T_N, x_N$. Since each T_i is the identifier for the corresponding turn x_i, a relation triple $(x_i, x_j, r_{i,j})$ with type $r_{i,j}$ can be serialized as $T_i, T_j : r_{i,j}$. Accordingly, we propose two prediction strategies: *Last Turn* (*D*^{2}*PSG-LT*) and *Full history* (*D*^{2}*PSG-FH*).

D^{2}PSG-LT. This strategy only focuses on the relations associated with the latest dialogue turn. For input $x_1, \ldots, x_i$, it only asks a model to predict one relation triple $T_i, T_j : r_{i,j}$, where $j < i$. For example in Fig. 3, only $T_2, T_0 : qap$ needs to be predicted.

D^{2}PSG-FH. This strategy requires predicting all discourse relations from each input $x_1, \ldots, x_i$. For example in Fig. 3, all relations, i.e. $T_1, T_0 : qap$ and $T_2, T_0 : qap$, are concatenated as the target sequence for prediction.

Compared with *D^{2}PSG-LT*, *D^{2}PSG-FH* may benefit from the partial predicted discourse relations. But that also brings error propagation.

B. Model

We use a T5 [1] model for sequence generation due to its strong generality. Similar to *Classifier-Hier* and *Classifier-Concat*, the T5 encoder is first adopted to encode dialogue history. Next, the T5 decoder is taken to perform discourse parsing by generating each linearized discourse relation triples in an autoregressive manner:

$$P(Y_i) = \texttt{T5} - \texttt{Dec}\left(\texttt{T5} - \texttt{Enc}(X), Y_{<i}\right), \tag{5}$$

where X indicates the current dialogue context, and Y represents the target token sequence of linearized discourse-relation triples. Our model is finetuned with standard cross-entropy loss:

$$\mathcal{L} = -\sum_{i=1}^{|Y|} \log P(Y_i). \tag{6}$$

As the T5 encoder and decoder have been jointly pretrained with large-scale self-supervised signals, their parameters are well initialized, and the decoder can well exploit the rich features from encoder via cross attention mechanism. Therefore, our model can quickly adapt to dialogue discourse parsing task with limited training data.

We apply constraint decoding to ensure that our model generates legal sequences under our policies. Particularly, it is required to each complete triple $T_i, T_j : r_{i,j}$, where $j < i$ and $r_{i,j}$ is a discourse relation. Under *D^{2}PSG-LT*, it is required to produce one complete triple with T_i being the marker of the latest turn. Under *D^{2}PSG-FH*, it is required to produce the same number of triples as the number of dialogue turns, and for each triple $T_i, T_j : r_{i,j}$, T_i needs to be the marker of the corresponding turn.

C. Leveraging Task Descriptions

Comparing with the classification-based systems, our model can better capture the semantic meanings of the discourse relations by generating their corresponding strings (e.g., "acknowledgement"), rather than treating them as independent categories of a classifier output space. Inspired by recent work on prompting [22], [23], we further leverage the descriptions of discourse relations to help our model better understand their semantic meanings. As shown in Fig. 3, we concatenate the descriptions of all discourse relations as additional model inputs. This can especially help our model on these relations whose corresponding strings are abbreviations (e.g., "qap" and "q-elab"), which are not directly understandable.

We simply the definitions from the annotation guidelines of *STAC* corpus [8] as task descriptions. Besides, we also add the example words mentioned in the guideline. For instance, the example words of the "acknowledgement" relation are *OK, Right, Right then, Good, Fine, etc.* More details can be found in Appendix.

TABLE I
EXPERIMENTS OF THE BASELINES USING DIFFERENT PLMS ON *MOLWENI*

Pretrained LM	Classifier-Hier		Classifier-Concat	
	Link	Link&Rel	Link	Link&Rel
RoBERTa-base	79.95	57.44	79.65	57.84
T5-base Enc.	80.26	57.44	80.04	57.44
RoBERTa-large	80.14	57.96	79.88	58.25
T5-large Enc.	80.10	57.71	82.73	58.95

V. EXPERIMENT

A. Setup

Datasets: We conduct experiments on two benchmark datasets: (*i*) **Molweni**. It is a multi-party dialogue corpus manually annotated based on Ubuntu Chat Corpus [24], which contains 9,000, 500 and 500 dialogues for training, development and testing, respectively. (*ii*) **STAC**. This dataset is collected from an online game. It is much smaller than *Molweni* and only contains 1,062 and 111 dialogues for training and testing, respectively.

Evaluation Metric: Following previous work, we evaluate our models and baselines with two scores: (*i*) *Link* F_1. It only measures whether the discourse link is correctly predicted. (*ii*) *Link&Rel* F_1. It is the **main** metric, measuring whether *both* the discourse link and the relation type are correctly predicted at the same time. Note that F_1 here denotes micro-averaged F_1 score.

Settings: We set T5.1.1 [1] with different model scales[3] as the backbone of our model and baselines. A batch size of 16/64/256 is selected for models with a T5-small/T5-base/T5-large encoder. All models are trained using Adam optimizer with linear scheduler and initial learning rate of 5e-5. As some extreme cases contain hundreds of utterances, all models take at most 20 latest utterances as inputs.

B. Baselines With Encoder-Only Pretrained LM

To make fair comparisons, we use T5-family models for all systems in later main experiments, which is different from previous efforts that leverage encoder-only pretrained LM. Therefore, we first conduct additional experiments for *Classifer-Hier* and *Classifier-Concat* using either a T5 encoder or a RoBERTa [25], a popular encoder-only pretrained LM. As shown in Table I, T5 encoder is quite competitive over RoBERTa across various model sizes, proving the fairness of our experiment settings.

C. Main Results

Table II compares our models with baselines and the previous approaches. All previous approaches (the first group) take a hierarchical encoder. Both *Hierarchical GRU* and *Structure Self-Aware* require whole dialogue content for classification, thus they are not applicable to online situations (e.g., online chatbot). On the other hand, our models (*D^{2}PSG-LT* and *D^{2}PSG-LT*) and baselines (*Classifier-Hier* and *Classifier-Concat*) analyze each ongoing dialogue given its partial content.

[3]We use the pretrained checkpoints from https://huggingface.co/models.

TABLE II
MAIN TEST RESULTS ON *MOLWENI* AND *STAC* BENCHMARKS. †;DENOTES MODELS OPERATING IN THE *OFFLINE* MANNER AND ⋆ REPRESENTS SOME LATEST WORK DURING THE SUBMISSION OF THIS PAPER. NOTE THAT [7] DOES NOT REPORT NUMBERS ON STANDARD TEST SPLIT, AND *CLASSIFIER-CONCAT* IS OUR IMPLEMENTATION OF [7] BASED ON OUR FRAMEWORK

Model	Pretrained LM	#Param.	Molweni		STAC	
			Link	Link&Rel	Link	Link&Rel
DeepSequential [9]	None	3M	76.80	54.03	71.58	53.77
Hierarchical GRU [11]†	RoBERTa-base	132M	79.70	55.90	75.30	56.90
Structure Self-Aware [10]†	ELECTRA-small	14M	81.63	58.54	73.48	57.31
⋆ SDDP [13]†	RoBERTa-base	140M	83.50	59.90	74.40	59.60
⋆ HG-MDP [12]†	BERT-base	144M	81.50	58.50	72.00	55.60
Classifier-Hier	T5-small Enc.	38M	78.79	56.51	71.73	55.19
	T5-base Enc.	112M	80.26	57.44	72.00	55.69
	T5-large Enc.	344M	80.10	57.71	71.24	55.85
Classifier-Concat	T5-small Enc.	35M	78.00	55.99	71.15	52.16
	T5-base Enc.	109M	80.04	57.44	72.37	56.02
	T5-large Enc.	341M	82.73	58.95	74.65	57.36
D^{2}PSG-LT	T5-small	77M	79.05	55.91	72.08	55.49
	T5-base	247M	80.51	57.31	75.07	59.29
	T5-large	783M	86.08	61.74	77.61	61.49
D^{2}PSG-FH	T5-small	77M	77.53	53.03	70.10	51.22
	T5-base	247M	80.29	54.95	72.22	55.23
	T5-large	783M	84.16	59.34	75.91	60.16
D^{2}PSG-LT w/ description	T5-small	77M	78.66	56.61	73.84	56.16
	T5-base	247M	82.17	58.25	76.25	59.39
	T5-large	783M	**87.07**	**62.01**	**78.40**	**62.77**

First, enlarging model size from T5-small to T5-large can generally improve all systems. However, the amount of improvement varies from 1.2 *Link&Rel* F_1 points for *Classifier-Hier* to almost 3.0 *Link&Rel* F_1 points for *Classifier-Concat* and more than 6.0 *Link&Rel* F_1 points for our models on *Molweni* test set. Since *Classifier-Hier* takes more randomly initialized parameters (classifiers and dialogue-encoder) than *Classifier-Concat* (only classifiers) and our models (none), this indicates the negative effect of using randomly initialized parameters. On the other hand, *Classifier-Hier* gives the best performances on both *Molweni* and *STAC* test sets under the T5-small model. This may explain why early neural models [9], [10], [11] tend to adopt a hierarchical encoder.

Second, with T5-large model as backbone, both *D^{2}PSG-FH* and *D^{2}PSG-LT* outperform previous SOTA systems and our baselines on the two benchmarks, showing the advantages of our sequence generation framework. While the *Link&Rel* F_1 scores of *D^{2}PSG-FH* and *D^{2}PSG-LT* are close under the T5-large model, their performance gaps are larger under a smaller model. This is because *D^{2}PSG-FH* suffers from more noise in historic predictions of discourse relations under a smaller model. We may expect another performance boost for *D^{2}PSG-FH* by using a larger pretrained model (e.g., T5-3B), while this is beyond our hardware budget at this time. Nevertheless, *D^{2}PSG-LT* can be a better choice over *D^{2}PSG-FH* under most currently affordable pretrained models.

Third, *D^{2}PSG-LT* using a T5-base model as the backbone significantly outperforms all baselines using a T5-large encoder on the *STAC* test set. On the other hand, it is slightly worse than the baselines on the *Molweni* test set. Since *STAC* contains much fewer training instances than *Molweni*, this indicates that our model is less data hungry. We conduct more analysis in Section V-D and Section V-E.

Finally, *D^{2}PSG-LT w/ description*, which concatenates relation-type descriptions with dialogue context as inputs, outperforms *D^{2}PSG-LT* no matter what pretrained model is used as the backbone. This demonstrates the usefulness of additional descriptions on our model for better understanding the semantic information of relation types.

D. Transfer Learning

Table III and IV show the results on domain transfer from *Molweni* to *STAC* and from *STAC* to *Molweni*, respectively. Compared with the in-domain results in Table II, the performances of all systems drop significantly due to domain shift (Ubuntu vs. Game). Generally, enlarging model size from T5-small to T5-large has relatively less benefits and can even hurt the performances of baseline systems, with *Classifier-Concat* being more robust than *Classifier-Hier*. On the other hand, the performances of our models keep increasing in most cases. This confirms the importance of avoiding randomly initialized parameters with a large-scale pretrained model. [11] explores several methods on target domain integration from both data and model perspectives. Though our models show inferior results on *Link* F_1, we still manage to significantly outperform their method on *Link&Rel* F_1, the main metric. Besides, our contributions are intuitively orthogonal to theirs.

Surprisingly, *SDDP* [13] shows strong performance in this setting by integrating theorems knowledge [26] and applying the

TABLE III
CROSS DOMAIN TRANSFER FROM *MOLWENI* TO *STAC*

Model	PLM	Molweni → STAC Link	Link&Rel
Classifier-Hier	T5-small	34.75	23.22
	T5-base	35.72	24.51
	T5-large	32.23	22.27
Classifier-Concat	T5-small	46.53	26.15
	T5-base	44.96	26.91
	T5-large	44.77	28.31
D^{2}PSG-LT	T5-small	43.16	25.70
	T5-base	44.04	25.85
	T5-large	45.09	29.47
D^{2}PSG-LT w/ description	T5-small	43.02	25.31
	T5-base	43.75	27.43
	T5-large	47.09	30.22
Hierarchical GRU [11]	RoBERTa-base	48.30	26.60
w/ domain integration	RoBERTa-base	50.50	28.90
⋆ SDDP [13]	RoBERTa-base	**50.60**	**31.60**

TABLE IV
CROSS DOMAIN TRANSFER FROM *STAC* TO *MOLWENI*

Model	PLM	STAC → Molweni Link	Link&Rel
Classifier-Hier	T5-small	57.87	33.82
	T5-base	54.68	34.57
	T5-large	46.38	29.07
Classifier-Concat	T5-small	62.14	34.54
	T5-base	59.96	34.75
	T5-large	58.70	35.32
D^{2}PSG-LT	T5-small	56.86	33.35
	T5-base	61.80	34.90
	T5-large	61.36	35.64
D^{2}PSG-LT w/ description	T5-small	58.02	32.94
	T5-base	61.16	35.62
	T5-large	61.77	36.47
Hierarchical GRU [11]	RoBERTa-base	60.70	31.50
w/ domain integration	RoBERTa-base	63.20	33.10
⋆ SDDP [13]	RoBERTa-base	**64.50**	**38.00**

maximum spanning tree decoding algorithm (MST, [8], [27]). We believe similar ideas may further benefit our model as well.

E. Performances on Few-Shot Learning

Table V show the system performances on *STAC* test set in low-resource settings, such as when only 10 (∼1%) and 100 (∼10%) dialogues are available for training. Using 10 dialogues for training, *Classifier-Hier* performs significantly worst than all other systems. Though *Classifier-Concat* is comparable with our models, it does not benefit much (1.0 *Link&Rel* F_1 point) from enlarging model size. Conversely, our models show highly competitive performances with all model sizes, and the performance gain can be nearly 5.0 *Link&Rel* F_1 points. This demonstrates that our models are less data hungry than baselines. Using 100 dialogues for training, all systems perform much better.

TABLE V
FEW SHOT LEARNING WITH 10 DIALOGUES (∼1%) AND 100 (∼10%) DIALOGUES FROM *STAC* SPLITTED BY DOUBLE LINES

Model	Pretrained LM	Link	Link&Rel
Classifier-Hier	T5-small	46.75	26.90
	T5-base	49.72	26.80
	T5-large	47.06	27.31
Classifier-Concat	T5-small	59.29	29.19
	T5-base	60.03	29.16
	T5-large	61.98	30.23
D^{2}PSG-LT	T5-small	59.87	28.48
	T5-base	59.03	28.47
	T5-large	64.50	33.30
D^{2}PSG-LT w/ description	T5-small	59.12	28.98
	T5-base	61.17	29.12
	T5-large	**64.87**	**33.46**
Classifier-Hier	T5-small	65.24	42.45
	T5-base	65.96	43.72
	T5-large	61.59	42.67
Classifier-Concat	T5-small	65.94	39.58
	T5-base	66.92	41.89
	T5-large	66.33	40.07
D^{2}PSG-LT	T5-small	66.50	41.90
	T5-base	69.56	45.71
	T5-large	74.03	50.84
D^{2}PSG-LT w/ description	T5-small	67.56	43.17
	T5-base	70.20	48.08
	T5-large	**74.20**	**51.94**

Our models again significantly outperform both baselines with T5-large, and they enjoy much more performance gains (nearly 9.0 *Link&Rel* F_1 points) from enlarging model size than the baselines (less than 1.0 *Link&Rel* F_1 point).

F. Performances on Long-Tail Cases

Fig. 4 analyzes the performances of multiple systems on the 16 relation types defined in *STAC*. As shown in the top sub-figure, these types are unevenly distributed with top 3 types and last 6 types covering 53% and 7.5% instances, respectively. This causes long tail issue. Both *D^{2}PSG-LT* and *D^{2}PSG-LT w/ desc.* outperform others for most long-tail relation types. Particularly, *Classifier-Hier*, *Classifier-Concat* and *D^{2}PSG-LT* achieve *Link&Rel* F_1 scores of 16.5%, 24.7% and 32.9% on the last 6 relation types. Besides, *D^{2}PSG-LT w/ desc.* is more advantageous than *D^{2}PSG-LT* across most types. Both results indicate the effectiveness of our model and adding task descriptions for handling rare instances.

G. Performances At Different Dialogue Turns

For a dialogue with more turns, it is more challenging because the dialogue context is more complex and discourse links need to be predicted from more utterance candidates. As shown in Fig. 5, we investigate our model and baselines at various dialogue turns on the *STAC* dataset, which contains many long conversations. For the first few turns, all models show competitive performance

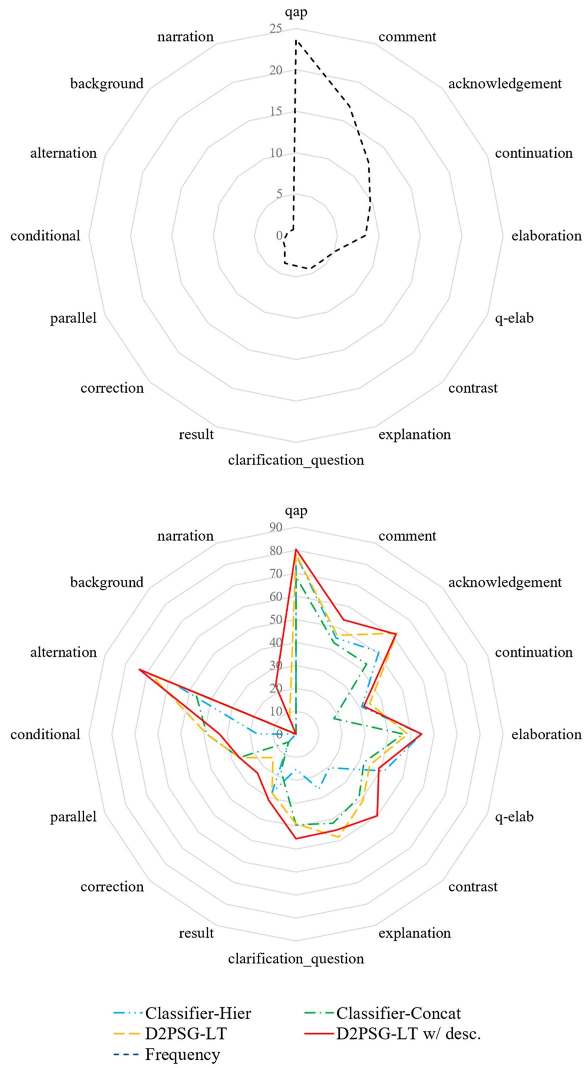

Fig. 4. Statistics on *STAC* across multiple relation types: frequency in training data (up) and *Link&Rel* F_1 scores (down). All models take T5-large as backbone.

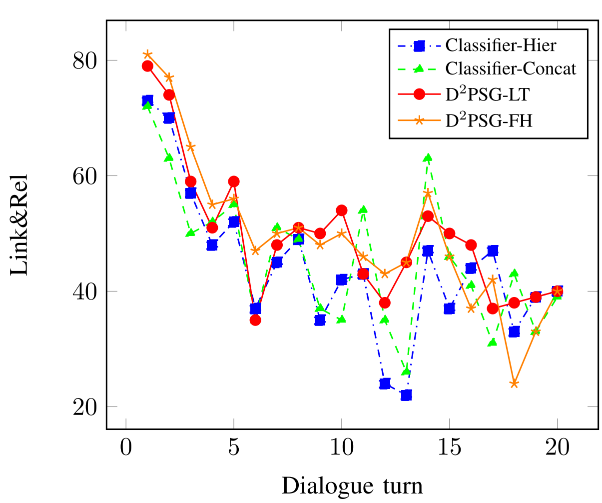

Fig. 5. Comparison of prediction accuracy between various models based on T5-large at different dialogue turns on *STAC* test set.

and our models perform slightly better. While, different models vary greatly after the 8-th turns and both *Classifier-Hier* and *Classifier-Concat* show intense fluctuation with dialogue turn increasing (e.g. 14-th vs. 15-th). Comparing with D^2*PSG-LT*, D^2*PSG-FH* has better performance within first few turns, while it shows inferior result with turn increasing. In particular, for the first 10 turns, D^2*PSG-FH* and D^2*PSG-LT* reach 61.49% and 60.26% points regarding *Link&Rel* F_1. However, for remaining turns, D^2*PSG-FH* is much worse than D^2*PSG-LT* (42.82% vs. 45.03%). This shows that D^2*PSG-FH* can benefit from partially predicted structure, but error propagation hurts more than the benefit for later turns.

H. Case Study

As shown in Table VI from the Appendix, we demonstrate a challenging example to help visualize the merits of our model. The oral conversation has 29 dialogue turns and contains many ellipses and coreferences, leading to great challenges for discourse parsers to correctly process this conversation. Generally, classification-based models perform worse than our models that are based on sequence generation. Besides, we notice that *Classifier-Concat* predicts more accurately than *Classifier-Hier* for the second half of the conversation. It confirms the advantage of using less randomly initialized parameters for better processing complex context. Compared with the baselines, our models not only perform better in overall but also are more accurate for these low-frequency relation types, such as "parallel", "correction" and "narration". For instance, both of our models successfully predict "(t13, t0, narration)". It is a long dependency relation across 13 dialogue turns and "narration" is a low-frequency relation type in the training set, which again shows the superiority of our approach.

VI. Related Work

A. Dialogue Discourse Parsing

Discourse parsing is a series of fundamental tasks, serving as the previous necessary step or additional feature inputs for various downstream tasks [2], [3], [4], [5], [6], [7]. In this work, we mainly focus on multi-party dialogue discourse parsing that aims to recognize the discourse relations among the utterances within one dialogue session. Since dialogues are usually organized differently from plain-text documents, several benchmarks [5], [8] have been proposed to accelerate this line of research.

Early attempts [8], [28] were devoted to improving decoding algorithms but only considered merely two involved utterances (local information) for predicting their relation. With the development of dialogue modeling, later studies [9], [10], [25] took the whole dialogue session (global information) into consideration for exploiting richer features. As illustrated in Section III, these studies can be generally sorted into two categories: *Hierarchical Encoder* and *Flat Encoder*. Most work belongs to the former one, using a hierarchical encoder consisting of token-level and sentence-level modules to encode each utterance and the whole dialogue session respectively. [9] sequentially

TABLE VI
AN EXAMPLE FROM STAC TEST SET, WHERE ALL MODELS ARE BASED ON T5-LARGE AND AN LINK / LINK&REL ERROR IS HIGHLIGHTED IN BLUE / PURPLE COLOUR

Dialogue	[t0] ztime: anyone want wheat? [t1] Shawnus: for? [t2] ztime: sheep? [t3] somdechn: Sheep? [t4] Shawnus: k [t5] ztime: yer sheep [t6] Shawnus: haha [t7] somdechn: 2 Wheat [t8] Shawnus: 1! [t9] Shawnus: undercut.. [t10] ztime: :-) [t11] somdechn: Goos one.. [t12] Shawnus: ruthless [t13] ztime: anyone else want wheat? [t14] somdechn: Yes you r.. [t15] somdechn: ok me [t16] somdechn: What do you want? [t17] somdechn: Sheep. [t18] ztime: do you have clay or wood? [t19] somdechn: I do have wood. [t20] somdechn: for 2 Wheats... [t21] ztime: ADDTIME [t22] ztime: OK [t23] somdechn: You are winning... [t24] somdechn: whay I'm tranding with you??? [t25] somdechn: Arrr.. [t26] Shawnus: haha [t27] ztime: na.. [t28] ztime: I'm stuck here...
Gold Query	(t1, t0, q-elab), (t2, t1, question-answer pair), (t3, t2, clarification question), (t4, t2, acknowledgement), (t5, t3, question-answer pair), (t6, t5, acknowledgement), (t7, t6, q-elab), (t8, t7, contrast), (t8, t5, q-elab), (t9, t8, elaboration), (t10, t8, comment), (t11, t8, comment), (t12, t8, comment), (t12, t11, elaboration), (t13, t0, narration), (t14, t12, acknowledgement), (t15, t13, question-answer pair), (t16, t13, q-elab), (t16, t15, q-elab), (t17, t16, elaboration), (t18, t16, q-elab), (t19, t18, question-answer pair), (t20, t19, elaboration), (t21, t18, continuation), (t22, t19, acknowledgement), (t23, t22, contrast), (t24, t23, clarification question), (t25, t24, comment), (t26, t25, acknowledgement), (t27, t23, correction), (t28, t27, elaboration)
Classifier-Hier	(t1, t0, q-elab), (t2, t1, q-elab), (t3, t2, q-elab), (t4, t2, question-answer pair), (t5, t3, question-answer pair), (t6, t5, comment), (t7, t4, elaboration), (t8, t5, comment), (t9, t8, elaboration), (t10, t9, comment), (t11, t8, comment), (t12, t11, comment), (t13, t0, continuation), (t14, t13, question-answer pair), (t15, t14, continuation), (t16, t15, q-elab), (t17, t16, question-answer pair), (t18, t16, q-elab), (t19, t18, question-answer pair), (t20, t17, elaboration), (t21, t19, result), (t22, t21, acknowledgement), (t23, t22, comment), (t24, t23, clarification question), (t25, t23, comment), (t26, t23, comment), (t27, t24, question-answer pair), (t28, t27, explanation)
Classifier-Concat	(t1, t0, q-elab), (t2, t1, q-elab), (t3, t2, q-elab), (t4, t3, question-answer pair), (t5, t3, question-answer pair), (t6, t5, comment), (t7, t5, elaboration), (t8, t7, continuation), (t9, t8, elaboration), (t10, t9, comment), (t11, t9, comment), (t12, t11, comment), (t13, t0, continuation), (t14, t13, question-answer pair), (t15, t14, continuation), (t16, t15, q-elab), (t17, t16, question-answer pair), (t18, t16, q-elab), (t19, t18, question-answer pair), (t20, t19, elaboration), (t21, t19, result), (t22, t19, acknowledgement), (t23, t22, comment), (t24, t23, clarification question), (t25, t24, comment), (t26, t25, comment), (t27, t24, question-answer pair), (t28, t27, explanation)
D^{2}PSG-LT	(t1, t0, q-elab), (t2, t1, question-answer pair), (t3, t2, q-elab), (t4, t2, question-answer pair), (t5, t3, question-answer pair), (t6, t5, comment), (t7, t5, q-elab), (t8, t7, correction), (t9, t8, explanation), (t10, t9, comment), (t11, t8, comment), (t12, t11, comment), (t13, t0, narration), (t14, t13, question-answer pair), (t15, t14, continuation), (t16, t15, continuation), (t17, t16, elaboration), (t18, t16, q-elab), (t19, t18, question-answer pair), (t20, t19, q-elab), (t21, t19, continuation), (t22, t19, acknowledgement), (t23, t22, result), (t24, t23, clarification question), (t25, t24, continuation), (t26, t25, comment), (t27, t25, comment), (t28, t27, elaboration)
D^{2}PSG-LT w/ description	(t1, t0, q-elab), (t2, t1, question-answer pair), (t3, t2, parallel), (t4, t2, question-answer pair), (t5, t3, question-answer pair), (t6, t5, comment), (t7, t5, q-elab), (t8, t7, correction), (t9, t8, explanation), (t10, t9, comment), (t11, t8, comment), (t12, t11, comment), (t13, t0, narration), (t14, t13, question-answer pair), (t15, t14, continuation), (t16, t15, continuation), (t17, t16, elaboration), (t18, t16, q-elab), (t19, t18, question-answer pair), (t20, t19, elaboration), (t21, t19, continuation), (t22, t19, acknowledgement), (t23, t22, result), (t24, t23, clarification question), (t25, t24, continuation), (t26, t25, comment), (t27, t25, comment), (t28, t27, elaboration)

predicted each relation and jointly considered previous predictions at each step. [10] proposed an edge-centric model based on Graph Transformer to directly learn features of each utterance pair. [11] was the first to explore cross-domain transfer between existing benchmarks. For the latter category, there is only one work [7] which directly feeds an entire session into a pretrained langugae model. Different from these studies, our work is the first attempt to investigate and tackle the curse of model scaling on this task. Besides, we study *online* setting, which is applicable to wider applications but is ignored by most current practices.

During the submission of this article, we notice some latest work that is worth discussing. [12] proposed a speaker-aware model that takes each speaker as a special node in their Graph Neural Network (GNN). As an important feature in this multi-party setup, integrating speaker information into the dialogue modeling is still worth exploring. [13] innovatively proposed a principled method by combining theorems [26], [27] and the latest practice. Based on RoBERTa-base [25], their model has outperformed previous efforts and even performs better than our best model in cross-domain settings. This inspires us to further enhance our method in the future via integrating their effective structured knowledge into our model, such as by introducing additional loss [29].

B. Modeling Various Tasks as Unified Sequence Generation

With the development of pretrained language models, researchers have extended the standard encode-only [30] architecture to decoder-only [31] and encoder-decoder [1], [32] architectures. This paves the way for solving various downstream tasks with one sequence generation process without adding new parameters. Particularly, there are several recent attempts that solve various tasks as sequence generation with a pretrained encoder-decoder model. For example, [33] adopted a pretrained BART [32] model to perform entity linking and document retrieval by generating the title of the entity or document token by token. [34] unified 4 information extraction tasks as sequence generation, which is then solved by a pretrained T5 [1]. Another line of research [35], [36], [37], [38] propose to integrate the task meta information into pretrained language models for low-source settings, where the meta information includes task definitions, annotation instructions, and even ontology descriptions.

TABLE VII
DESCRIPTIONS FOR 16 RELATION TYPES

qap	an answer to the question
elaboration	provide more information about what was said (for instance, first, second)
acknowledgement	an understanding or acceptance of what was said (ok, right, good, fine)
clarification_question	an question to eliminate or prevent misunderstanding, confusion or ambiguity
result	the effect of a cause (so)
comment	provide an opinion or evaluation of what was said
q-elab	a follow-up question to get more information to answer a first question
explanation	explain why, or give the cause of what happened (because)
contrast	(but, however, on the other hand, nevertheless, while)
parallel	(too, also)
alternation	(or)
conditional	(if then)
correction	–
background	–
narration	–
continuation	–

Inspired by these studies, we are the first to formulate dialogue discourse parsing as a sequence generation problem and further leverage task descriptions to help model better understand the semantic meaning of each relation type.

VII. CONCLUSION

We formulated multi-party dialogue discourse parsing as a sequence generation task, which was then solved by a well-pretrained encoder-decoder model. Since our model does not take any randomly initialized parameters, it is more effective and less data hungry than previous SOTA systems and our carefully designed baselines using randomly initialized classifiers. We introduced two strategies, i.e. D^2*PSG-LT* and D^2*PSG-FH*, to linearize discourse relations into a sequence, and we explored adding relation-type descriptions to help model understand their semantic information. Experiment results on two benchmarks validated the effectiveness of our approach. Besides, we further demonstrated the robustness of our model with zero-shot, few-shot evaluations and other in-depth analyses.

APPENDIX
DESCRIPTIONS

To get the descriptions, we consult the annotation guidelines of *STAC* corpus, where each relation type is directly defined or explained with several examples. As shown in Table VII, there are 16 relation types in total. We simplify these definitions and copy the example words which are then enclosed between parentheses as our relation type descriptions. As some relation types are nontrivial, we leave their descriptions empty. In future work, we will study to apply more accurate descriptions with extra human efforts to our model.

ACKNOWLEDGMENT

The authors thank reviewers for their insightful comments and also thank editors for their every effort.

REFERENCES

[1] C. Raffel et al., “Exploring the limits of transfer learning with a unified text-to-text transformer,” *J. Mach. Learn. Res.*, vol. 21, pp. 1–67, 2020.
[2] Q. Jia, Y. Liu, S. Ren, K. Zhu, and H. Tang, “Multi-turn response selection using dialogue dependency relations,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 1911–1920.
[3] J. Chen and D. Yang, “Structure-aware abstractive conversation summarization via discourse and action graphs,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2021, pp. 1380–1391.
[4] X. Feng, X. Feng, B. Qin, and X. Geng, “Dialogue discourse-aware graph model and data augmentation for meeting summarization,” in *Proc. 30th Int. Joint Conf. Artif. Intell.*, Z.-H. Zhou, Ed., Aug. 2021, pp. 3808–3814. [Online]. Available: https://doi.org/10.24963/ijcai.2021/524
[5] J. Li et al., “Molweni: A challenge multiparty dialogues-based machine reading comprehension dataset with discourse structure,” in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 2642–2652.
[6] S. Ouyang, Z. Zhang, and H. Zhao, “Dialogue graph modeling for conversational machine reading,” *Findings Assoc. Comput. Linguistics: ACL-IJCNLP*, pp. 3158–3169, 2021.
[7] Y. He, Z. Zhang, and H. Zhao, “Multi-tasking dialogue comprehension with discourse parsing,” in *Proc. 35th Pacific Asia Conf. Lang., Inf. Comput.*, 2021, pp. 69–79.
[8] S. Afantenos, É. Kow, N. Asher, and J. Perret, “Discourse parsing for multi-party chat dialogues,” in *Proc. 2015 Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 928–937.
[9] Z. Shi and M. Huang, “A deep sequential model for discourse parsing on multi-party dialogues,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 01, 2019, pp. 7007–7014.
[10] A. Wang et al., “A structure self-aware model for discourse parsing on multi-party dialogues,” in *Proc. 30th Int. Joint Conf. Artif. Intell.*, Z.-H. Zhou, Ed., Aug. 2021, pp. 3943–3949. [Online]. Available: https://doi.org/10.24963/ijcai.2021/543
[11] Z. Liu and N. Chen, “Improving multi-party dialogue discourse parsing via domain integration,” in *Proc. 2nd Workshop Comput. Approaches Discourse*, 2021, pp. 122–127.

[12] J. Li, M. Liu, Y. Wang, D. Zhang, and B. Qin, "A speaker-aware multiparty dialogue discourse parser with heterogeneous graph neural network," *Cogn. Syst. Res.*, vol. 79, pp. 15–23, 2023.

[13] T.-C. Chi and A. Rudnicky, "Structured dialogue discourse parsing," in *Proc. 23rd Annu. Meeting Special Int. Group Discourse Dialogue*, 2022, pp. 325–335.

[14] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[15] I. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 3776–3783.

[16] I. Serban et al., "A hierarchical latent variable encoder-decoder model for generating dialogues," in *Proc. AAAI Conf. Artif. Intell.*, 2017, vol. 31. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/10983

[17] C. Xing, Y. Wu, W. Wu, Y. Huang, and M. Zhou, "Hierarchical recurrent attention network for response generation," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 5610–5617.

[18] J. Zhu, J. Li, M. Zhu, L. Qian, M. Zhang, and G. Zhou, "Modeling graph structure in transformer for better AMR-to-text generation," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 5459–5468.

[19] D. Cai and W. Lam, "Graph transformer for graph-to-sequence learning," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 7464–7471.

[20] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[21] G. Hinton et al., "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[22] T. Brown et al., "Language models are few-shot learners," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 1877–1901.

[23] C.-H. Lee, H. Cheng, and M. Ostendorf, "Dialogue state tracking with a language model using schema-driven prompting," in *Proc. 2021 Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 4937–4949.

[24] R. Lowe, N. Pow, I. V. Serban, and J. Pineau, "The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems," in *Proc. 16th Annu. Meeting Special Int. Group Discourse Dialogue*, 2015, pp. 285–294.

[25] Y. Liu et al., "Roberta: A robustly optimized bert pretraining approach," 2019, *arXiv:1907.11692*.

[26] W. T. Tutte and W. T. Tutte, *Graph Theory*, vol. 21. Cambridge, U.K.: Cambridge Univ. Press, 2001.

[27] P. Muller, S. Afantenos, P. Denis, and N. Asher, "Constrained decoding for text-level discourse parsing," in *Proc. 24th Int. Conf. Comput. Linguistics*, 2012, pp. 1883–1900.

[28] J. Perret, S. Afantenos, N. Asher, and M. Morey, "Integer linear programming for discourse parsing," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, San Diego, California, 2016, pp. 99–109. [Online]. Available: https://aclanthology.org/N16-1013

[29] L. Song et al., "Structural information preserving for graph-to-text generation," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7987–7998.

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[31] A. Radford et al., "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019. [Online]. Available: https://openai.com/research/gpt-2-6-month-follow-up

[32] M. Lewis et al., "Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880.

[33] N. D. Cao, G. Izacard, S. Riedel, and F. Petroni, "Autoregressive entity retrieval," in *Int. Conf. Learn. Representations*, 2020.

[34] Y. Lu et al., "Unified structure generation for universal information extraction," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers*, 2022, pp. 5755–5772.

[35] A. Rastogi, X. Zang, S. Sunkara, R. Gupta, and P. Khaitan, "Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 8689–8696.

[36] F. Mi, Y. Wang, and Y. Li, "Cins: Comprehensive instruction for few-shot learning in task-oriented dialog systems," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 10, 2022, pp. 11076–11084.

[37] Q. Luo, L. Liu, Y. Lin, and W. Zhang, "Don't miss the labels: Label-semantic augmented meta-learner for few-shot text classification," in *Findings Assoc. Comput. Linguistics*, 2021, pp. 2773–2782.

[38] A. Mueller et al., "Label semantic aware pre-training for few-shot text classification," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers)*, 2022, pp. 8318–8334.