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.

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. 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 35022202006059. 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: wangle@stu.xmu.edu.cn; yao0010@xmu.edu.cn; jsu@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

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

2In 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 (dmn, 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...
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 $u_i^{(0)}$ of each utterance $x_i$:

$$u_i^{(0)} = \text{MeanPool} (\text{T5} - \text{Enc}(x_i))$$

A Graph Transformer of $T$ layers is then used to update the initial utterance representations (e.g., $u_i^{(0)}$) with more global information. Following previous work, each input graph is fully connected with the utterances as its nodes. The label (e.g., $e_{ij}$) 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:

$$u_i^{(t+1)} = \sum_{j=1}^{N} \alpha_{ij} (u_j^{(t)} W^V + e_{ij} W^F),$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'=1}^{N} \exp(e_{ij}')} ,$$

$$e_{ij} = \left( u_i^{(t)} W^Q \right) \left( u_j^{(t)} W^K + e_{ij} W^R \right)^\top \sqrt{d_i} ,$$

Authorized licensed use limited to: Xiamen University. Downloaded on November 01,2023 at 08:36:57 UTC from IEEE Xplore. Restrictions apply.
where $e_{i,j}$ is the embedding vector for the edge connecting $x_i$ and $x_j$, $u_i^{(i)}$ is the $i$-th utterance representation at the $t$-th layer, and all the $W^z$ are model parameters.

Finally, we define the feature vector $F_{i,j}$ between $x_i$ and $x_j$ as $[u_i^{(0)}; u_i^{(T)}; u_j^{(0)}; u_j^{(T)}]$, 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:

$$L_i = L_{i,link} + L_{i,rel},$$

$$L_{i,link} = - \sum_{j=1}^{i-1} \log P_{i,link}(x_i^+ = x_j \mid F_{i,j}),$$

$$L_{i,rel} = - \log P_{i,rel}(r_{i,j}^+ \mid F_{i,j}), \quad (3)$$

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 $x_i$ is the gold discourse target of $x_j$. Note that if $x_i$ does not depend on any preceding utterance (e.g., being the first utterance), then $x_i^+ = x_i$ and $r_{i,j}^+ = \text{none}$.

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:

$$u_1, \ldots, u_i = \text{T5} - \text{Enc}(x_1[\text{SEP}] \ldots x_i[\text{SEP}]) \quad (4)$$

As the next step, we follow [7] to get the feature vector $F_{i,j} = (u_i, u_j, u_i - u_j, u_i \cdot u_j)$, 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 $T_2, T_0 : \text{gap}$ 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. $T_1, T_0 : \text{gap}$ and $T_2, T_0 : \text{gap}$, 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:

$$P(Y_i) = \text{T5} - \text{Dec} (\text{T5} - \text{Enc}(X), Y_{<i}),$$  

(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:

$$L = -\sum_{i=1}^{\vert Y \vert} \log P(Y_i).$$  

(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.

### 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 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 Classifier-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-FH) and baselines (Classifier-Hier and Classifier-Concat) analyze each ongoing dialogue given its partial content.

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

### 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 w/ desc. 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
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
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 language 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 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

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

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. $D^2PSG$-$LT$ and $D^2PSG$-$FH$, 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


