Structured Dialogue Discourse Parsing

Ta-Chung Chi
Language Technologies Institute
Carnegie Mellon University
tachungc@andrew.cmu.edu

Alexander I. Rudnicky
Language Technologies Institute
Carnegie Mellon University
air@cs.cmu.edu

Abstract

Dialogue discourse parsing aims to uncover the internal structure of a multi-participant conversation by finding all the discourse *links* and corresponding *relations*. Previous work either treats this task as a series of independent multiple-choice problems, in which the link existence and relations are decoded separately, or the encoding is restricted to only local interaction, ignoring the holistic structural information. In contrast, we propose a principled method that improves upon previous work from two perspectives: encoding and decoding. From the encoding side, we perform structured encoding on the adjacency matrix followed by the matrix-tree learning algorithm, where all discourse links and relations in the dialogue are jointly optimized based on latent tree-level distribution. From the decoding side, we perform structured inference using the modified Chiu-Liu-Edmonds algorithm, which explicitly generates the labeled multi-root non-projective spanning tree that best captures the discourse structure. In addition, unlike in previous work, we do not rely on hand-crafted features; this improves the model's robustness. Experiments show that our method achieves new state-of-the-art, surpassing the previous model by 2.3 on STAC and 1.5 on Molweni (F1 scores). [1]

1 Introduction

Discourse parsing is a series of tasks that consist of elementary discourse unit (EDU) segmentation, relation directionality classification (optional), and relation type classification between EDUs (Jurafsky and Martin, 2021). It serves as the first step of many downstream applications (Meyer and Popescu-Belis, 2012; Jansen et al., 2014; Narasimhan and Barzilay, 2015; Bhatia et al., 2015; Ji et al., 2016; Asher et al., 2016; Ji and Smith, 2017; Li et al., 2020a), and it can be categorized into three major discourse formalisms: RST (Mann and Thompson, 1988), PDTB (Prasad et al., 2008), and SDRT (Lascarides and Asher, 2008) styles. Considering that SDRT-style formalism is used to label the STAC (Asher et al., 2016) and Molweni (Li et al., 2020a) dialogue corpora and the increasing importance of dialogue discourse parsers trained on them (Ouyang et al., 2021; Feng et al., 2021; Jia et al., 2020; Chen and Yang, 2021), we focus on designing an SDRT-style dialogue discourse parser using the two corpora in this work. Figure 1 presents an example of a dialogue session in the STAC corpus (Asher et al., 2016) annotated with its discourse structure. The annotation is often encoded in two components: *links* and *relations*. The goal of a dialogue discourse parser is to extract them accurately at the same time.

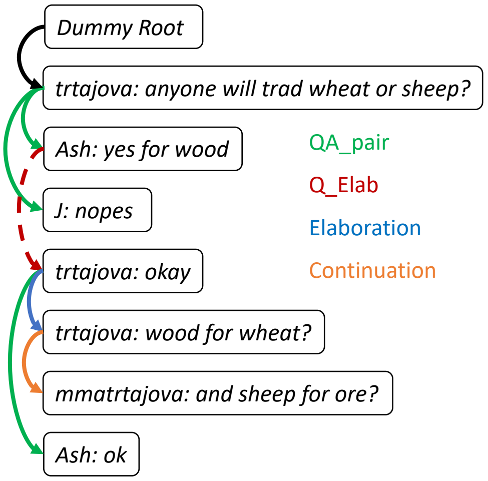
Dummy Root
trtajova: anyone will trad wheat or sheep?
Ash: yes for wood
J: nopes
trtajova: okay
trtajova: wood for wheat?
mmatrtajova: and sheep for ore?
Ash: ok
QA_pair
Q_Elab
Elaboration
Continuation

Figure 1: This is an example dialogue session. The ultimate goal of a dialogue discourse parser is to predict all the links (arrows) and relations (color of arrows) shown in this figure. Note that the Q_Elab arrow (dashed) can cross the QA_pair one, making it non-projective.

One straightforward solution to this problem is to transform the parsing structure into a series of local pairwise link prediction problems. In other words, the model is expected to compute some

[1] Code released at https://github.com/chijames/structured_dialogue_discourse_parsing.

Proceedings of the SIGdial 2022 Conference, pages 325–335
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022 ©2022 Association for Computational Linguistics

Models	Encoding	Decoding	Link & Relation Prediction	Use Feature
MST (2015)	*local, edge-wise*	*partial MST*	*separate*	Y
ILP (2016)	*local, edge-wise*	*ILP*	*separate*	Y
Deep-Seq. (2019)	*global, two-staged*	*indp. multiple choice*	*separate*	Y
Struct-Aware (2021)	*global, fully-connected*	*indp. multiple choice*	*separate*	Y
Hierarchical (2021)	*hierarchical*	*indp. multiple choice*	*separate*	N
This Work	*global, structured*	*full MST*	*joint*	N

Table 1: This is the comparison between different dialogue discourse parsers. Our method is designed with structured encoding and decoding processes. Furthermore, the links and relations are learned and predicted jointly. Finally, our method does not rely on human-designed features, hence enjoys better robustness.

local potentials between each pair of utterances, and predict the relation type of that link if it exists. However, this formulation does not take the global structural information into account, leading to inferior parsing performance.

In contrast to previous work, our core observation is that by adding a dummy utterance at the beginning of the dialogue, the overall structure closely resembles a **labeled multi-root non-projective spanning tree**. In light of this observation, we propose a principled dialogue discoursing parser that encodes structural inductive biases during training and inference.

The essential elements of our method are the novel structured parameterization of the adjacency matrix, the directed version of matrix-tree theorem (Tutte, 1984; Koo et al., 2007), and the modified directed spanning tree inference algorithm. To the best of our knowledge, this is the first time that the labeled multi-root non-projective spanning tree is applied to the analysis of dialogue discourse structure. In summary, the contributions of this paper are:

- We propose a principled method for the dialogue discourse parsing task, where structural inductive biases for both encoding and decoding processes are introduced.
- We jointly predict discourse *links* and *relations* in a unified space.
- We propose a padding method that allows batchwise variable-length determinant calculation.
- Experimental results demonstrate state-of-the-art discoursing parsing performance on two datasets.

2 Task Background

We are given a dialogue session D and the links and relations between pairs of utterances labeled using the 17 discourse relations defined in Asher et al. (2016). All the utterances, links, and relations constitute a graph $G(V, E, R)$, where V represents the set of utterances, E represents the links connecting them, and R represents the edge labels. The goal of a discourse parser is to predict E and R given V.

There are five existing dialogue discourse parsers to the best of our knowledge (Afantenos et al., 2015; Perret et al., 2016; Shi and Huang, 2019; Wang et al., 2021; Liu and Chen, 2021). We compare them against each other in detail in the following subsections and provide a summary in Table 1.

2.1 Encoding

Afantenos et al. (2015); Perret et al. (2016) use a MaxEnt (Ratnaparkhi, 1997) model to parameterize local pairwise scores between utterance pairs. Therefore, global and contextual information are not taken into account during the encoding process. Liu and Chen (2021) improve upon them by using a hierarchical encoder that models the contextual information. Shi and Huang (2019) inject more structural information by first predicting all the links, followed by a global structured encoding module. However, the predicted links are discrete, making this two-staged solution not end-to-end trainable. To connect the two stages, Wang et al. (2021) instead use a fully connected graph between all utterances. While being fully end-to-end, useful structured bias is not encoded anymore. Based on the drawbacks of previous parsers, we propose a fully end-to-end encoder while maintaining structured information at the same time.

2.2 Decoding

Shi and Huang (2019); Wang et al. (2021); Liu and Chen (2021) treat the links and relations decoding tasks as a series of independent multiple-choice problems. In other words, the existence of one link has nothing to do with other links. In contrast, Perret et al. (2016) find the structure by solving an integer linear programming problem, but it needs a set of complicated human-designed decoding constraints. Afantenos et al. (2015) is the closest approach to this work, where they run the maximum spanning tree decoding algorithm on the predicted edges only to find the tree structure (links). However, the relations are not jointly decoded. Instead, we run the modified spanning tree decoding algorithm on the unified link and relation space.

2.3 Link and Relation Prediction

All previous work treat the prediction of links and relations as a two-stage process. That is, they first predict the existence of a link, and the relation is predicted only if the link exists. This decouples the joint learning of links and relations. We mitigate this issue by unifying the prediction space of links and relations, making it a three-dimensional tensor.

2.4 Feature Usage

Finally, all previous work execpt (Liu and Chen, 2021) utilize some hand-crafted features. To name a few, they explicitly model if two utterances are spoken by the same *speaker*, or if they belong to the same *turn*. These features are useful but also make the baseline parsers deeply coupled with them, which might limit the parsing performance if applied to a new dataset. For example, if the new dataset is a transcript of a teleconference or radio exchange, it is likely that we only have the utterances recorded as it is expensive and hard to obtain all the speaker and turn information. In contrast, since our model does not rely on such explicitly modeled feature, the performance drop is less than the ones that use them when the speaker and turn information are removed.

3 Structure Formulation

The graph G defined in § 2 can theoretically be any directed acyclic graph, which is generally difficult to optimize. Fortunately, we find that by discarding only a small fraction of the edges, which is 6% for the STAC corpus and 0% for the molweni corpus, we can recover a spanning tree-like structure that permits efficient learning and structure inference. For the nodes having more than one parent, we keep only the latest one.[2] In addition, for dangling utterances that do not have any parents, we connect them to the dummy root utterance, so we are in fact optimizing a *multi-root* tree during training time. Finally, note that our tree structure allows different links to *cross* each other (Figure 1) and each edge also has a relation *label*, $G(V, E, R)$ is a labeled directed multi-root non-projective spanning tree, which is referred to as tree for conciseness hereinafter [3].

Several questions naturally arise:

- How to parametrize the tree? We will model the pairwise potential scores by an adjacency matrix, where a cell represents the relevance score of a pair of utterances. See § 4.1.
- How to learn the correct tree? We calculate the probability of the correct tree among all possible trees encoded by the adjacency matrix, and that probability is maximized. This is similar to softmax attention using trees as basic units instead of tokens. See § 4.2.
- How to perform inference? Given the learned three-dimensional adjacency matrix, we can run the modified maximum spanning tree induction algorithm to induce the tree structure. See § 4.4.

4 Proposed Framework

4.1 Model Parameterization

Given n utterances (aka EDUs) $\{U_i\}_{i=1}^{n}$ in a dialogue session D, we define a *discourse pair* in D as a 3-tuple $(h, m, r), h < m, r \in [1, 17]$ where $h \in [0 \ldots n]$ is the index of the parent utterance, $m \in [1 \ldots n]$ is the index of the children utterance, and r is one of the 17 relations. Note that we add a special *root* utterance $h = 0$ to be the shared pseudo parent for the first utterances. Note that this root utterance can be chosen arbitrarily, and we use the utterance “This is the start of a dialogue” in this work.

To parameterize the tree, we first model the d-dimensional pairwise representation between a pair of utterances. This can be expressed compactly by a 3-order tensor, which is an adjacency matrix with each element $V_{h,m}$ being a d-dimensional feature vector, hence $V \in \mathbb{R}^{(n+1)\times(n+1)\times d}$. Each $V_{h,m}$ is calculated using a BERT (Devlin et al., 2019) model as the encoder. BERT takes a pair of utterances as input, and a special [CLS] token

[2] This strategy is adopted by all baselines as well.

[3] There are four types of spanning trees investigated in the dependency parsing domain (McDonald et al., 2005; Koo et al., 2007)

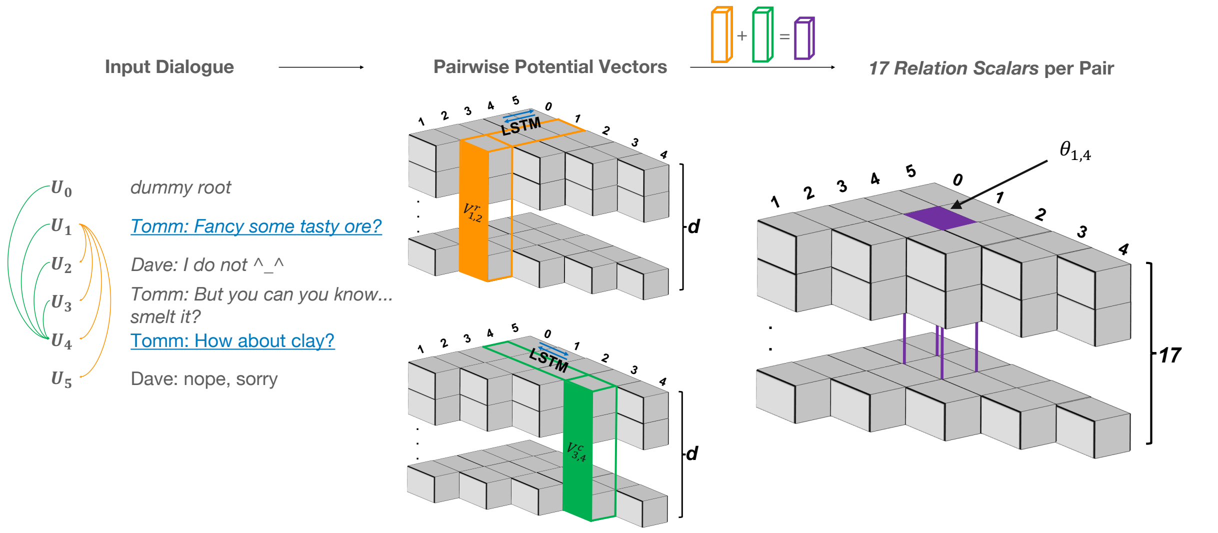

Figure 2: The contextual encoding process. Row numbers from 0 to 4 represent h, and column numbers from 1 to 5 represents m. In this example, the 1-st utterance can connect to the 2-nd, 3-rd or 5-th utterances when predicting the $V_{1,4}$ cell. This is represented by the orange rectangles. Similarly, the 4-th utterance can have 0, 1, or 3-rd utterances as its parent, represented by the green rectangles. We use two LSTMs for two directions (orange and green). Finally, we add the contextualized vectors together to get the purple vector Linear$(V^r_{1,4} + V^c_{1,4}) = \theta_{1,4}$.

is prepended before the concatenation of the two utterances. The representation of the [CLS] token is further used to calculate d-dimensional pairwise representation:

$$V_{h,m} = \text{BERT}_{CLS}(U_h, U_m) \tag{1}$$

One immediate drawback of eq. (1) is that the pairwise scores are calculated independently. We alleviate this issue by using a bidirectional LSTM to encode the contextual information. For the h-th row, we obtain the hidden states for all timestep t:

$$\{V^{\text{r}}_{h,t}\}^n_{t=h+1} = \text{LSTM}(\{V_{h,t}\}^n_{t=h+1}) \tag{2}$$

The underlying idea is that to accurately decide if U_h should point to U_m, we should collect the information of connecting U_h to all the other utterances that appear later chronologically. Similarly, for the m-th column, all the hidden states are:

$$\{V^{\text{c}}_{t,m}\}^{m-1}_{t=0} = \text{LSTM}(\{V_{t,m}\}^{m-1}_{t=0}) \tag{3}$$

The final context-aware potential score is:

$$\tilde{V}_{h,m} = V^{\text{r}}_{h,m} + V^{\text{c}}_{h,m} \tag{4}$$

$\tilde{V} \in \mathbb{R}^{(n+1)\times(n+1)\times 2d}$. Every pairwise score is now aware of neighboring pairs. It still remains to convert $\tilde{V}$ to individual score of a discourse pair. We do so by simply passing $\tilde{V}$ through a linear transformation layer:

$$\theta_{h,m} = \text{Linear}(\tilde{V}) \tag{5}$$

where $\theta \in \mathbb{R}^{(n+1)\times(n+1)\times 17}$. Note that there is no activation function after the linear layer since we assume θ to be in log space. Another important property of θ is that it is a strictly *upper-triangular* matrix due to the $h < m$ constraint. In practice, this can be enforced by setting the lower triangular and diagonal elements to -inf. We illustrate the overall idea in Figure 2.

4.2 Learning the Tree

The parameterization we define in eq. (5) still does not impose any structural constraints. Based on our conclusion in § 3, we would like to impose a non-projective multi-root spanning tree constraint in the learning and inference process. We define a tree T to be the collection of discourse pairs $\{(h, m, r)\}$. We use $\mathcal{T}(D)$ to denote all possible trees of a dialogue session D. During learning, the reference tree structure $\bar{T} \in \mathcal{T}(D)$ is given. If the score of $\bar{T}$ and the summation of the scores of all trees in $\mathcal{T}(D)$ is tractable, we can obtain the probability of $\bar{T}$ and optimize it using gradient descent. The challenge lies in the exponentially many candidates of $\mathcal{T}(D)$, which is computationally infeasible to naively enumerate. Fortunately, we will see that the Matrix-Tree Theorem (Tutte, 1984; Koo et al., 2007) permits efficient calculation of the summation we need.

4.2.1 Matrix-Tree Theorem

Before we dive into the details of the Matrix-Tree Theorem, we have to give enough credits to Tutte

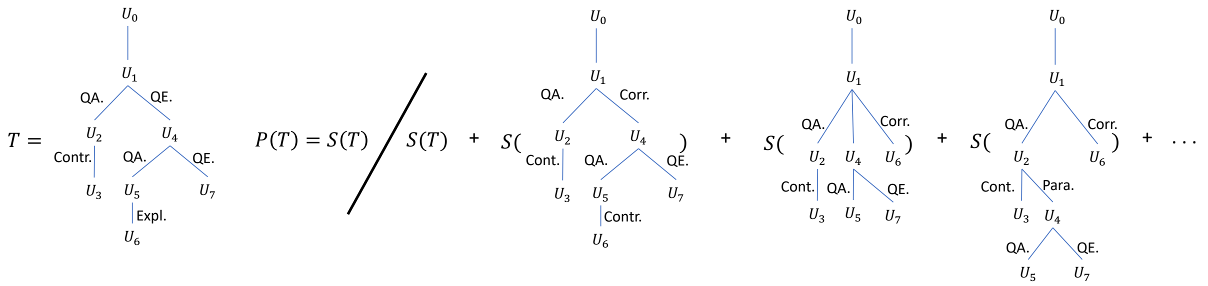

Figure 3: This is the graphical illustration of of how we perform tree-structured learning. Note that we are summing the scores of all possible labeled trees as the denominator, which is eq. (11).

(1984); Koo et al. (2007) as we are applying their proposed theorem. The probability of the reference tree $\bar{T}$, which is to be optimized, can be defined as:

$$\mathbb{P}(\bar{T}) = \frac{s(\bar{T})}{Z(\theta)}, \quad Z(\theta) = \sum_{T \in \mathcal{T}(D)} s(T) \tag{6}$$

$Z(\theta)$ is also known as the partition function. The numerator $s(T)$ of any tree T is defined to be:

$$s(T) = \prod_{(h,m,r) \in T} \exp(\theta_{h,m,r}), \tag{7}$$

With this definition, the score is merely the product of corresponding cells in $\exp(\theta)$ (θ from eq. (5)).

Next, we need to find an efficient way to compute the partition function as there are exponentially many candidate trees. The first step is to calculate the exponential matrix $A_{h,m,r}$ from eq. (5):

$$A_{h,m,r}(\theta) = \begin{cases} 0, & \text{if } h \geq m \\ \exp(\theta_{h,m,r}), & \text{otherwise} \end{cases} \tag{8}$$

Note that the first $0 = \exp(-inf)$ condition applies to all $h \geq m$ cells as θ is an upper-triangular matrix described earlier. To account for edge labels, we have to marginalize r out :

$$A_{h,m}(\theta) = \sum_r A_{h,m,r}(\theta) \tag{9}$$

Now, we are ready to calculate $Z(\theta)$. The first step is to calculate the graph Laplacian matrix, which is the difference between the degree matrix and adjacency matrix:

$$L_{h,m}(\theta) = \begin{cases} \sum_{i'=1}^{n} A_{i',m}(\theta), & \text{if } h = m \\ -A_{h,m}(\theta), & \text{otherwise} \end{cases} \tag{10}$$

Then the minor[4] $L^{(0,0)}(\theta)$ is equal to the sum of the weights of all directed spanning trees rooted at the dummy root utterance (Tutte, 1984):

$$Z(\theta) = \det(\hat{L}), \tag{11}$$

whre $\hat{L}$ is defined to be the submatrix constructed by removing the first row and column from L. The computational complexity of eq. (11) is determined by the determinant operation, which is $O(n^3)$. While the cubic time complexity might seem scary at the first glance, it does not incur significant computational overhead in our experiments, where the time to compute the determinant is negligible (< 1%) compared to BERT encoding in eq. (1).

4.2.2 Efficient GPU Implementation

The equations derived so far work well for a single training instance. However, it becomes problematic if we want to perform batchwise training on GPUs, which was not addressed in Koo et al. (2007). The main challenge is the variable-length padding. In particular, we have to calculate batchwise determinants in eq. (11) with different sizes of $\hat{L}$. The naive option is to pad the extra rows and columns with zeros. Unfortunately, this would result in a singular matrix and give erroneous partition results. To circumvent the padding issue, we can use the cofactor expansion formula. Concretely, all the diagonal elements of the padding part should be 1, while others should be 0. We illustrate the padding strategy in Figure 4. Note that this strategy holds whether the size of $\hat{L}$ is odd or even.

4.3 Optimization of Tree

Since we are given the reference tree $\bar{T}$, we can directly maximize the log probability of eq. (6) using any gradient-descent based algorithms, which is also equivalent to minimizing the KL-divergence between the predicted and reference tree distributions.

[4] A minor $L^{(x,y)}$ is the determinant of a submatrix constructed by removing the x-th row and y-th column of L.

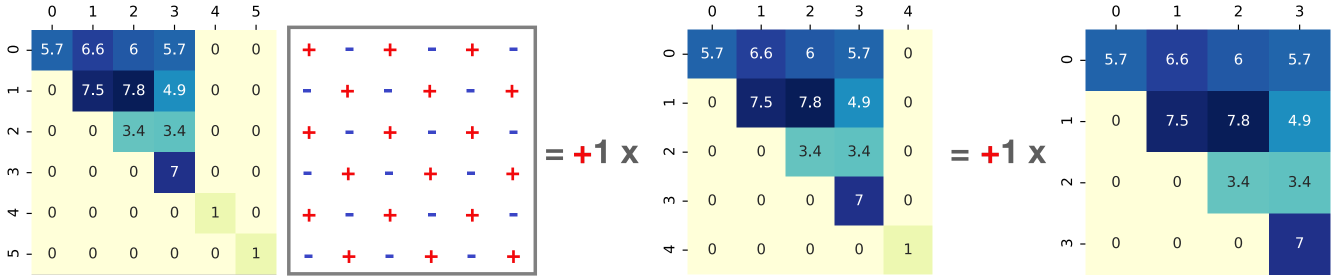

Figure 4: This is an efficient padding for calculating batch determinant. The original 4×4 matrix is expanded to a 6×6 (leftmost) one for padding. Note that the last two diagonal elements are all ones. The second matrix encodes the coefficients for multiplying sub-matrices. After a series of cofactor expansions, we can see that the determinant of the padded 6×6 matrix is equivalent to the original unpadded 4×4 matrix.

4.4 Inference of Tree

There is a well-known algorithm - Chiu-Liu-Edmonds (CLE) (Edmonds, 1967; Chu, 1965) that can find the directed spanning tree $\tilde{T}$ with maximum weight given $A(\theta)$ derived in eq. (8). However, we cannot directly apply the CLE algorithm as the original version does not accept labeled trees. To solve this problem, we have to first pick the highest-scoring relation for each edge $A'_{h,m} = \max_r A_{h,m,r}$ to get $A' \in \mathbb{R}^{(n+1)\times(n+1)}$. Now we can feed this standard form into the CLE algorithm: $\tilde{T} = \text{CLE}(A')$. The correctness of this approach can be proved easily by contradiction: suppose the optimal tree includes one edge that is not the highest-scoring one among $\{A_{h,m,r}\}_{r=1}^{17}$, we can always substitute that edge with the highest-scoring one to get a better tree (contradiction). Note that for a pair of utterances, we only allow one direction of link (θ is strictly upper triangular) so the CLE algorithm in fact degenerates to its undirected version known as Prim/Kruskal's algorithm (Prim, 1957; Kruskal, 1956).

5 Experiments

5.1 Datasets

There are two datasets for us to train the discourse parser, one of which is the STAC (Asher et al., 2016) corpus, which is a multi-party dialogue corpus collected from an online game, and the other is the Ubuntu IRC corpus (Li et al., 2020a), which compiles technical discussions about Linux. The differences between these two datasets were analyzed in Liu and Chen (2021), where the takeaway messages are: 1) there is no significant difference in their average EDU numbers, 2) the lexical distributions are significantly different sharing only a small portion of common tokens, 3) relation distributions are similar.

5.2 Hyperparameters

Following Liu and Chen (2021), we use the Roberta-Base uncased pretrained checkpoint for a fair comparison. The max utterance length is set to 28. The initial learning rate is set to 2e-5 with a linear decay to 0 for 4 epochs. The batch size is 4. The first 10% of training steps is the warmup stage. For all baselines using large pretrained models, we always use the same model checkpoint and tune the learning rate and batch size for them for a fair comparison.

5.3 Metrics

We follow the baselines to use two metrics for evaluation:

- Unlabeled Attachment Score (UAS): We only care about the existence of a discourse link. In other words, discourse relations do not affect the results. (Also known as Link F1 score)
- Labeled Attachment Score (LAS): It is much harder as it requires both discourse links and relations to be correct. We focus primarily on this metric since it is more informative and is used in downstream applications. (Also known as Link & Rel F1 score)

5.4 Main Results

We present the results in Table 2. The left part of the table focuses on in-domain training and testing, which is the standard setting. Bearing in mind that discourse parsers are often used as the first stage of downstream applications, we follow (Liu and Chen, 2021) to benchmark the performance of all parsers in the cross-domain setting. Note that this is an extremely challenging setting as the domains are completely different (gaming vs Linux technical forum).

	STAC / STAC		MOL / MOL		STAC / MOL		MOL / STAC	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
MST (Afantenos et al., 2015)	69.6	52.1	69.0	48.7	61.5	24.0	**60.5**	14.8
ILP (Perret et al., 2016)	69.0	53.1	67.3	48.3	57.0	24.1	60.4	14.5
Deep-Seq. (Shi and Huang, 2019)	73.2	54.4	76.1	53.3	53.5	21.6	42.7	15.7
Hierarchical (Liu and Chen, 2021)	73.1	57.1	80.1	56.1	60.1	32.1	48.9	26.8
Struct-Aware (Wang et al., 2021)	73.4	57.3	81.6	58.4	57.0	32.9	44.7	26.1
This Work	**74.4**	**59.6**	**83.5**	**59.9**	**64.5**	**38.0**	50.6	**31.6**

Table 2: STAC / MOL means the training dataset is STAC and the testing dataset is MOL. LAS is the harder setting used for downstream applications. Results are the average of three runs. Note that speaker information is still used in this set of experiments, except our parser does not need to model their relations explicitly as described in § 2.4.

In-domain We first take a look at the in-domain results. Our proposed parser is the best among all parsers, surpassing the previous state-of-the-art by 2.3 on STAC and 1.5 (F1 scores) on Molweni under the LAS setting. The trend is similar for the UAS setting. We also want to highlight the improved performance can NOT be attributed to using a pretrained language model as *Struct-Aware* and *Hierarchical* (Liu and Chen, 2021) both utilize the same or comparable pretrained model.

Cross-domain We shift gear to the cross-domain setting where the parser is trained on one dataset and tested on the other (Liu and Chen, 2021). We can see that our parser is the best under the LAS setting, substantially outperforming the best candidates by 5.1 on STAC/MOL and 4.8 points on MOL/STAC. However, the best-performing model under the UAS setting is the oldest model (Afantenos et al., 2015; Perret et al., 2016). This can be explained by the inclination of a large pretrained model to overfit on the training domain, which was corroborated by Liu and Chen (2021) as well. Readers might wonder why the same phenomenon does not happen under the UAS setting of STAC/MOL, and the speculated reason is STAC has a much larger linguistic diversity (Liu and Chen, 2021), thereby alleviating the model overfitting issue. In other words, we might want to train the dialogue discourse parser on a linguistically diverse dataset if the goal is domain generalization.

5.5 Additional Analyses

Dialogue Length Robustness We hypothesize that our parser is likely to perform better when the dialogue becomes longer, and the reason is that our parser models the overall dialogue structure using tree distributions. This lowers the burden of the parser to predict long-range links. We focus on the in-domain setting and plot the results in Figure 5. As we can see, the performance of our parser drops less than baselines when the dialogue becomes longer, highlighting the benefit of global structured learning and inference.

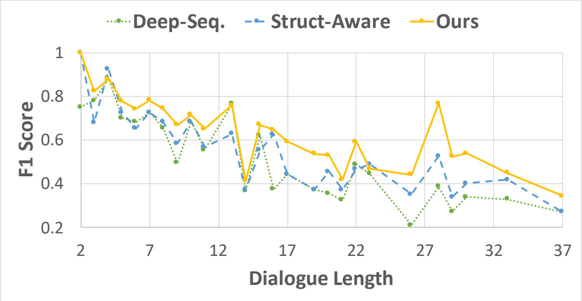

Figure 5: Parsing performance w.r.t dialogue lengths. As we can see, the performance difference is larger when the dialogue becomes longer, demonstrating the length robustness of our parser.

Relation Performance Breakdown In order to know what kinds of relations benefit the most from our proposed parser, we count the number of correct relation predictions and plot them in Figure 6 and 7.[5] The baseline parser we compare with is the Hierarchical model (Liu and Chen, 2021) as it can be viewed as the non-structured version of our parser with the same pretrained model backbone. We can see that our parser outperforms the baseline on certain relations like *Comment* and *QA pair* on STAC and *QA pair* and *Clarification Question* on Molweni. However, there is still a large room for improvement as demonstrated by the gap between our parser and the ground truth. Another observation is that both parsers struggle to predict

[5] Note that this implies the link predictions of these correct relations are also correct.

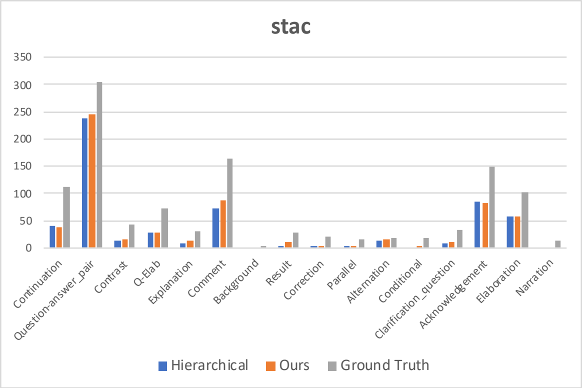

Figure 6: STAC relation performance breakdown.

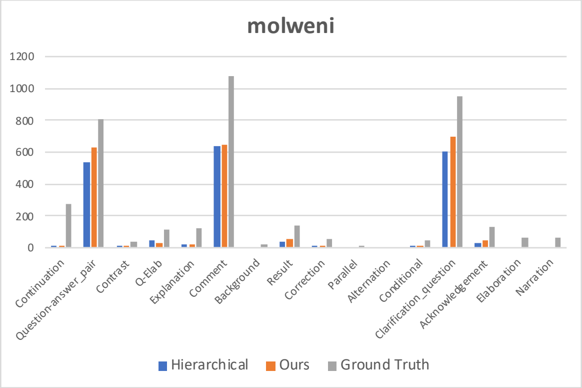

Figure 7: Molweni relation performance breakdown.

low-resource relations, marking an important direction for future work.

Speaker and Turn Feature Robustness We experiment with removing speaker and turn information used in baselines. The performance drop (LAS of STAC) of our parser (59.6 → 54.4) is less than that of the best baseline (Struct-Aware) (57.3 → 47.8), demonstrating the robustness of our parser.

6 Related Work

Discourse Parsing As discussed in the Introduction section, there are three types of discourse parsing formalisms: RST (Mann and Thompson, 1988), PDTB (Prasad et al., 2008), and SDRT (Lascarides and Asher, 2008; Asher et al., 2016). For the first two tasks, there are transition-based (Li et al., 2014; Braud et al., 2017; Yu et al., 2018) and CKY-based methods (Joty et al., 2015; Li et al., 2016; Liu and Lapata, 2017) in the literature. In this work, we assume that the EDUs are already given. In practice, there are papers working on segmenting EDUs (Subba and Di Eugenio, 2007; Li et al., 2018) before feeding them to the discourse parser.

Dialogue Disentanglement Clustering utterances in a conversation into threads is studied extensively by previous work (Shen et al., 2006; Elsner and Charniak, 2008; Wang and Oard, 2009; Elsner and Charniak, 2011; Jiang et al., 2018; Kummerfeld et al., 2019; Zhu et al., 2020; Li et al., 2020b; Yu and Joty, 2020). They predict the *reply-to* links independently and run a connected component algorithm to construct the threads. This is similar to the UAS setting in this work.

Structured Learning Algorithms Natural language is highly structured suggesting that the introduction of structural bias will facilitate learning. Previous work have studied dependency-tree like structures extensively (Koo et al., 2007; McDonald et al., 2005; McDonald and Satta, 2007; Niculae et al., 2018; Paulus et al., 2020). Several works propose to incorporate such inductive bias into intermediate layers of modern NLP models (Kim et al., 2017; Chen et al., 2017; Liu and Lapata, 2018; Choi et al., 2018). In our work, the induced structure is not only implicitly learned, it is also used to directly decode the labeled tree structure, which is our ultimate goal.

Dependency Parsing Our work can also be viewed as extending token-level dependency parsing (Mel'cuk et al., 1988; Koo et al., 2007; Smith and Eisner, 2008; Koo and Collins, 2010; Chen and Manning, 2014; Dozat and Manning, 2017; Qi et al., 2018; Choi and Palmer, 2011) to utterance-level. Another important difference is that our tree is labeled, which means we have to additionally predict the type of tree edges.

7 Conclusion

In this paper, we propose a principled method for dialogue discourse parsing. From the encoding side, we introduce a structurally-encoded adjacency matrix followed by the matrix-tree theorem, which is used to holistically model all utterances as a tree. From the decoding side, we apply the modified CLE algorithm for maximum spanning tree induction. Our method achieves state-of-the-art performance on two benchmark datasets. We also benchmark the cross-domain parser performance, and find our parser performs the best in the most-commonly used and harder LAS setting. We believe that the techniques described in this work pave the way for more structured analyses of dialogue and interesting research problems in the field of dialogue discourse parsing.

References

Stergos Afantenos, Eric Kow, Nicholas Asher, and Jérémy Perret. 2015. Discourse parsing for multiparty chat dialogues. Association for Computational Linguistics (ACL).

Nicholas Asher, Julie Hunter, Mathieu Morey, Farah Benamara, and Stergos Afantenos. 2016. Discourse structure and dialogue acts in multiparty dialogue: the stac corpus. In *10th International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2721–2727.

Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from RST discourse parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2212–2218, Lisbon, Portugal. Association for Computational Linguistics.

Chloé Braud, Maximin Coavoux, and Anders Søgaard. 2017. Cross-lingual RST discourse parsing. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 292–304, Valencia, Spain. Association for Computational Linguistics.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 2017. Improved neural machine translation with a syntax-aware encoder and decoder. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1936–1945, Vancouver, Canada. Association for Computational Linguistics.

Jiaao Chen and Diyi Yang. 2021. Structure-aware abstractive conversation summarization via discourse and action graphs. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1380–1391, Online. Association for Computational Linguistics.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2018. Learning to compose task-specific tree structures. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jinho D Choi and Martha Palmer. 2011. Getting the most out of transition-based dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 687–692.

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

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.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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

Micha Elsner and Eugene Charniak. 2008. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of ACL-08: HLT*, pages 834–842.

Micha Elsner and Eugene Charniak. 2011. Disentangling chat with local coherence models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1179–1189.

Xiachong Feng, Xiaocheng Feng, Bing Qin, and Xinwei Geng. 2021. Dialogue discourse-aware graph model and data augmentation for meeting summarization. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3808–3814. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Peter Jansen, Mihai Surdeanu, and Peter Clark. 2014. Discourse complements lexical semantics for nonfactoid answer reranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 977–986.

Yangfeng Ji, Gholamreza Haffari, and Jacob Eisenstein. 2016. A latent variable recurrent neural network for discourse-driven language models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 332–342, San Diego, California. Association for Computational Linguistics.

Yangfeng Ji and Noah A. Smith. 2017. Neural discourse structure for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1005, Vancouver, Canada. Association for Computational Linguistics.

Qi Jia, Yizhu Liu, Siyu Ren, Kenny Zhu, and Haifeng Tang. 2020. Multi-turn response selection using dialogue dependency relations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1911–1920, Online. Association for Computational Linguistics.

Jyun-Yu Jiang, Francine Chen, Yan-Ying Chen, and Wei Wang. 2018. Learning to disentangle interleaved conversational threads with a siamese hierarchical network and similarity ranking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1812–1822.

Shafiq Joty, Giuseppe Carenini, and Raymond T Ng. 2015. Codra: A novel discriminative framework for rhetorical analysis. *Computational Linguistics*, 41(3):385–435.

Daniel Jurafsky and James H. Martin. 2021. *22*, 3 edition.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. 2017. Structured attention networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Terry Koo and Michael Collins. 2010. Efficient third-order dependency parsers. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1–11.

Terry Koo, Amir Globerson, Xavier Carreras, and Michael Collins. 2007. Structured prediction models via the matrix-tree theorem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 141–150.

Joseph B Kruskal. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50.

Jonathan K. Kummerfeld, Sai R. Gouravajhala, Joseph J. Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C Polymenakos, and Walter Lasecki. 2019. A large-scale corpus for conversation disentanglement. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3846–3856, Florence, Italy. Association for Computational Linguistics.

Alex Lascarides and Nicholas Asher. 2008. Segmented discourse representation theory: Dynamic semantics with discourse structure. In *Computing meaning*, pages 87–124. Springer.

Jiaqi Li, Ming Liu, Min-Yen Kan, Zihao Zheng, Zekun Wang, Wenqiang Lei, Ting Liu, and Bing Qin. 2020a. 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.

Jing Li, Aixin Sun, and Shafiq R Joty. 2018. Segbot: A generic neural text segmentation model with pointer network. In *IJCAI*, pages 4166–4172.

Qi Li, Tianshi Li, and Baobao Chang. 2016. Discourse parsing with attention-based hierarchical neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 362–371.

Sujian Li, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014. Text-level discourse dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25–35.

Tianda Li, Jia-Chen Gu, Xiaodan Zhu, Quan Liu, Zhen-Hua Ling, Zhiming Su, and Si Wei. 2020b. Dialbert: A hierarchical pre-trained model for conversation disentanglement. *arXiv preprint arXiv:2004.03760*.

Yang Liu and Mirella Lapata. 2017. Learning contextually informed representations for linear-time discourse parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1289–1298.

Yang Liu and Mirella Lapata. 2018. Learning structured text representations. *Transactions of the Association for Computational Linguistics*, 6:63–75.

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.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 523–530.

Ryan McDonald and Giorgio Satta. 2007. On the complexity of non-projective data-driven dependency parsing. In *Proceedings of the Tenth International Conference on Parsing Technologies*, pages 121–132.

Igor Aleksandrovic Mel'cuk et al. 1988. *Dependency syntax: theory and practice*. SUNY press.

Thomas Meyer and Andrei Popescu-Belis. 2012. Using sense-labeled discourse connectives for statistical machine translation. In *Proceedings of the EACL2012 Workshop on Hybrid Approaches to Machine Translation (HyTra)*, CONF.

Karthik Narasimhan and Regina Barzilay. 2015. Machine comprehension with discourse relations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1253–1262.

Vlad Niculae, Andre Martins, Mathieu Blondel, and Claire Cardie. 2018. Sparsemap: Differentiable sparse structured inference. In *International Conference on Machine Learning*, pages 3799–3808. PMLR.

Siru Ouyang, Zhuosheng Zhang, and Hai Zhao. 2021. Dialogue graph modeling for conversational machine reading. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3158–3169, Online. Association for Computational Linguistics.

Max B. Paulus, Dami Choi, Daniel Tarlow, Andreas Krause, and Chris J. Maddison. 2020. Gradient estimation with stochastic softmax tricks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Robert Clay Prim. 1957. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal Dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium. Association for Computational Linguistics.

Adwait Ratnaparkhi. 1997. A simple introduction to maximum entropy models for natural language processing. *IRCS Technical Reports Series*, page 81.

Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. 2006. Thread detection in dynamic text message streams. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 35–42.

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.

David A Smith and Jason Eisner. 2008. Dependency parsing by belief propagation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 145–156.

Rajen Subba and Barbara Di Eugenio. 2007. Automatic discourse segmentation using neural networks. In *Proc. of the 11th Workshop on the Semantics and Pragmatics of Dialogue*, pages 189–190.

WT Tutte. 1984. Graph theory, encyclopedia of mathematics and it applications.

Ante Wang, Linfeng Song, Hui Jiang, Shaopeng Lai, Junfeng Yao, Min Zhang, and Jinsong Su. 2021. A structure self-aware model for discourse parsing on multi-party dialogues. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3943–3949. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Lidan Wang and Douglas W Oard. 2009. Context-based message expansion for disentanglement of interleaved text conversations. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 200–208. Citeseer.

Nan Yu, Meishan Zhang, and Guohong Fu. 2018. Transition-based neural rst parsing with implicit syntax features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 559–570.

Tao Yu and Shafiq Joty. 2020. Online conversation disentanglement with pointer networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6321–6330, Online. Association for Computational Linguistics.

Henghui Zhu, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. Who did they respond to? conversation structure modeling using masked hierarchical transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9741–9748.