A Deep Sequential Model for Discourse Parsing on Multi-Party Dialogues

Zhouxing Shi, Minlie Huang*
Dept. of Computer Science & Technology, Tsinghua University, Beijing 100084, China
Institute for Artificial Intelligence, Tsinghua University (THUAI), China
Beijing National Research Center for Information Science and Technology, China
zhouxingshichn@gmail.com; aihuang@tsinghua.edu.cn

Abstract

Discourse structures are beneficial for various NLP tasks such as dialogue understanding, question answering, sentiment analysis, and so on. This paper presents a deep sequential model for parsing discourse dependency structures of multi-party dialogues. The proposed model aims to construct a discourse dependency tree by predicting dependency relations and constructing the discourse structure jointly and alternately. It makes a sequential scan of the *Elementary Discourse Units (EDUs)*[1] in a dialogue. For each EDU, the model decides to which previous EDU the current one should link and what the corresponding relation type is. The predicted link and relation type are then used to build the discourse structure incrementally with a structured encoder. During link prediction and relation classification, the model utilizes not only *local* information that represents the concerned EDUs, but also *global* information that encodes the EDU sequence and the discourse structure that is already built at the current step. Experiments show that the proposed model outperforms all the state-of-the-art baselines.

Introduction

Discourse parsing is to identify relations between discourse units and to discover the discourse structure that the units form (Li, Li, and Chang 2016). Previous studies have shown that discourse structures are beneficial for various NLP tasks, including dialogue understanding (Asher et al. 2016; Takanobu et al. 2018), question answering (Verberne et al. 2007), information retrieval (Seo, Croft, and Smith 2009), and sentiment analysis (Cambria et al. 2013; Bhatia, Ji, and Eisenstein 2015).

Many approaches have been proposed for discourse parsing based on Rhetorical Structure Theory (RST) (Mann and Thompson 1988). However, RST is designed for written text and only allows discourse relations to appear between adjacent discourse units, and thus is inapplicable for multi-party dialogues (Afantenos et al. 2015) since multi-party dialogue data have more complex discourse structures in nature. RST is *constituency-based*, where related adjacent discourse units are merged to form larger units recursively, resulting in a hierarchical tree structure (Li, Li, and Hovy 2014). By contrast, *dependency-based* structures, where EDUs are directly linked without forming upper-level structures, are more applicable for multi-party dialogues. It is because multi-party dialogues have immediate relations between non-adjacent discourse units and the discourse structures are generally non-projective[2] (Morey, Muller, and Asher 2018). Therefore, the focus of this paper is on parsing *dependency structures* of multi-party dialogues. Figure 1 shows an example of a multi-party dialogue and its dependency structure, where three speakers (A, B, C) are conversing in an online game.

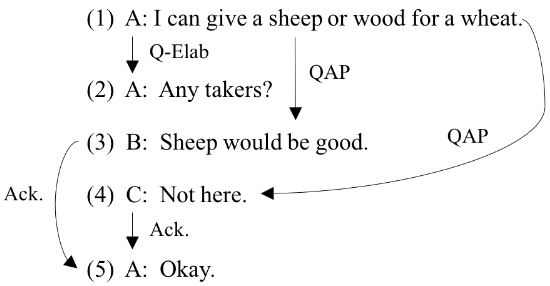

(1) A: I can give a sheep or wood for a wheat.
Q-Elab
QAP
(2) A: Any takers?
(3) B: Sheep would be good.
QAP
Ack.
(4) C: Not here.
Ack.
(5) A: Okay.

Figure 1: A multi-party dialogue example with its discourse structure from the STAC Corpus (Asher et al. 2016), where "Q-Elab" is short for "Question-Elaboration", "QAP" for "Question-Answer-Pair", and "Ack." for "Acknowledgement".

Prior state-of-the-art approaches for discourse dependency parsing commonly adopt a pipeline framework: first estimating the local probability of the dependency relation between each combination of two EDUs, and then constructing a discourse structure with decoding algorithms such as maximum spanning tree or integer linear programming

*Corresponding author: Minlie Huang.

[1] A discourse can be segmented into clause-level units called *Elementary Discourse Units (EDUs)* which are the most fundamental discourse units in discourse parsing. Following previous work such as (Li et al. 2014; Li, Li, and Hovy 2014), we also assume that EDU segmentations are preprocessed.

[2] A discourse structure is *non-projective* if it is impossible to draw the relations on the same side without crossing (McDonald et al. 2005). As the non-projective example shown in Figure 1, 1 → 4 and 3 → 5 have to be drawn on two sides to avoid crossing.

(Muller et al. 2012; Li et al. 2014; Afantenos et al. 2015; Perret et al. 2016), based on the estimated probabilities. However, these approaches have two drawbacks. **First**, the probability estimation of each dependency relation between two EDUs only relies on the local information of these two considered EDUs. **Second**, dependency prediction and discourse structure construction are separated in two stages, thereby dependency prediction cannot utilize the information from the predicted discourse structure for better dependency parsing, and in return, worse dependency prediction degrades the construction of the discourse structure.

To address these drawbacks, we propose a deep sequential model for discourse parsing on multi-party dialogues. This model constructs a discourse structure incrementally by predicting dependency relations and building the structure jointly and alternately. It makes a sequential scan of the EDUs in a dialogue. For each EDU, the model decides to which previous EDU the current one should link and what the relation type is. Such dependency prediction relies on not only local information that encodes the two concerned EDUs, but also global information that encodes the EDU sequence and the discourse structure that is already built at the current step. The predicted link and relation type, in return, are used to build the structure incrementally with a structured encoder. In this manner, the model predicts dependency relations and constructs the discourse structure jointly and alternately.

In summary, we make the following contributions:

- We propose a deep sequential model for discourse parsing on multi-party dialogues. The model predicts dependency relations and constructs a discourse structure jointly and alternately.
- We devise a prediction module that fully utilizes local information that encodes the concerned units, and also global information that encodes the EDU sequence and the currently constructed structure.
- We devise a structured encoder for representing structured global information, and propose a *speaker highlighting mechanism* to utilize speaker information and enhance dialogue understanding.

Related Work

Most previous work for discourse parsing is based on Penn Discourse TreeBank (PDTB) (Prasad et al. 2007) or Rhetorical Structure Theory Discourse TreeBank (RST-DT) (Mann and Thompson 1988). PDTB focuses on shallow discourse relations but ignores the overall discourse structure (Yang and Li 2018), while in this paper we aim to parse discourse structures. As for RST, there have been many approaches including transition-based methods (Braud, Coavoux, and Søgaard 2017; Wang, Li, and Wang 2017; Yu, Zhang, and Fu 2018) and those involving CYK-like algorithms (Joty, Carenini, and Ng 2015; Li, Li, and Chang 2016; Liu and Lapata 2017) or greedy bottom-up algorithms (Feng and Hirst 2014). However, constituency-based RST does not allow non-adjacent relations, which makes it inapplicable for multi-party dialogues. By contrast, in this paper, we aim to parse non-projective dependency structures, where dependency relations can appear between non-adjacent EDUs.

There have been some approaches proposed for parsing discourse dependency structures in two stages. These approaches first predict the local probability of dependency relation for each possible combination of EDU pairs, and then apply a decoding algorithm to construct the final structure. (Muller et al. 2012; Li et al. 2014; Afantenos et al. 2015) used Maximum Spanning Trees (MST) to construct a dependency tree, and (Muller et al. 2012) also attempted A^* algorithm but did not achieve better performance than MST. (Perret et al. 2016) further used Integer Linear Programming (ILP) to construct a dependency graph. However, these approaches predict the probability of a dependency relation only with the local information of the two considered EDUs, while the constructed structure is not involved. By contrast, our sequential model predicts dependency relations and constructs the discourse structure jointly and alternately, and utilizes the currently constructed structure in dependency prediction.

Although transition-based approaches for discourse dependency parsing have been proposed by (Jia et al. 2018a; 2018b) which also construct dependency structures incrementally, they still underperform the approach using MST by (Li et al. 2014). It is because these transition-based local approaches do not investigate other possible links when predicting a dependency relation as argued by (Jia et al. 2018b), and they are limited to predict projective structures. Therefore, these approaches are inapplicable for multi-party dialogues. By contrast, our sequential model predicts the parent of each EDU in the dependency tree by comparing all its preceding EDUs, and it can predict non-projective structures which are necessary for multi-party dialogues.

Moreover, state-of-the-art approaches for discourse dependency parsing as mentioned above still rely on handcrafted features or external parsers. Neural networks have recently been widely applied in various NLP tasks, including RST discourse parsing (Li, Li, and Chang 2016; Braud, Coavoux, and Søgaard 2017) and dialogue act recognition (Kumar et al. 2018; Chen et al. 2018). And (Jia et al. 2018a; 2018b) also applied neural networks in their transition-based dependency parsing models. In this paper, we adopt hierarchical Gated Recurrent Unit (GRU) (Cho et al. 2014) encoders to compute discourse representations.

Methodology

Problem Definition

We formulate the discourse dependency parsing problem on a multi-party dialogue as follows: given a dialogue that has been segmented into a sequence of EDUs $u_1, u_2, \cdots, u_n$, together with an additional dummy root u_0[3], the goal is to predict dependency links and the corresponding relation types $\{(u_j, u_i, r_{ji}) | j \neq i\}$ between the EDUs, where (u_j, u_i, r_{ji}) stands for a link of relation type r_{ji} from u_j to u_i. The predicted dependency relations should constitute a

[3] The dummy root is for the convenience of problem definition following (Li et al. 2014).

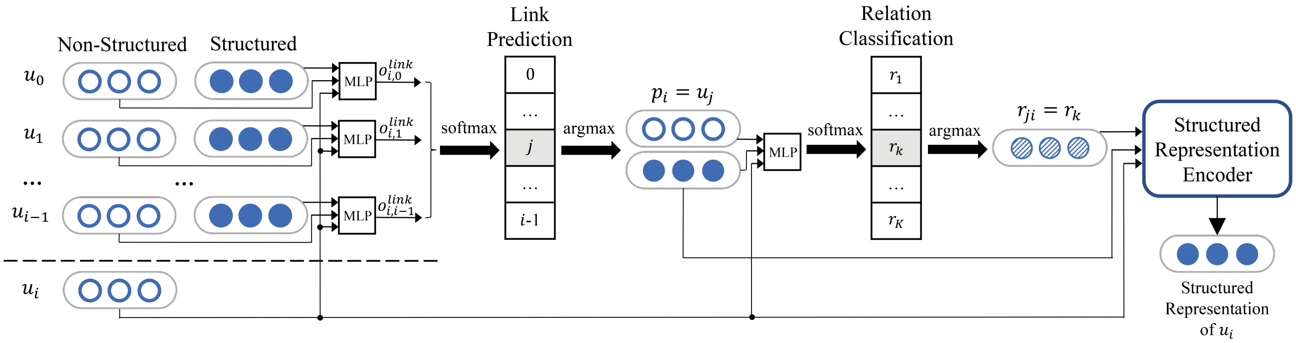


Figure 2: Illustration of the model which consists of modules for link prediction, relation classification, and structured representation encoding. For the current EDU u_i, link prediction estimates a distribution over its preceding EDUs, relation classification estimates a distribution over relation types, and the structured encoder updates the structured representation of u_i using representations of u_i and p_i and the embedding of the predicted relation type r_{ji}. Non-structured representation encoding is performed before the prediction process and is omitted from the illustration.

Directed Acyclic Graph (DAG) and there should be no relation linked to u_0.

The discourse structure predicted by our model is a dependency tree, which is a special type of DAG[4]. The model makes a sequential scan of the EDUs $u_1, u_2, \cdots, u_n$. For the current EDU u_i, the model predicts a dependency link by estimating a probability distribution as follows:

$$\mathcal{P}(u_j|u_i, \mathcal{T}_i, 0 \le j \le i-1) \qquad (1)$$

where $\mathcal{T}_i = \{(u_l, u_k, r_{lk})|0 \le l < k \le i-1\}$ is the set of dependency relations that are already predicted before the current step i. This is so-called *link prediction* in our model. Similarly, the model predicts the relation type for a predicted link $u_j \to u_i (j < i)$ with the following distribution:

$$\mathcal{P}(r_{ji}|u_j \to u_i, \mathcal{T}_i) \qquad (2)$$

where $r_{ji} \in \{r_1, r_2, \cdots, r_K\}$, $r_k (1 \le k \le K)$ is a relation type and K is the number of relation types. This is so-called *relation classification*.

Model Overview

Our model first computes the non-structured representations of the EDUs with hierarchical Gated Recurrent Unit (GRU) (Cho et al. 2014) encoders. These non-structured representations are used for predicting dependency relations and encoding structured representations. Next, the model makes a sequential scan of the EDUs and has the following three steps as illustrated in Figure 2 when it handles EDU u_i:

1. **Link prediction**: predict the parent node p_i of EDU u_i with a link predictor which utilizes not only non-structured representations, but also structured representations that encode the predicted structure before u_i. Specifically, we compute a score between the current EDU u_i and each linking candidate $u_j (j < i)$ with an MLP. These scores are then normalized to a distribution over the previous EDUs $\{u_0, u_1, \cdots, u_{i-1}\}$ with $softmax$, from which we can take the linked EDU with the largest probability.
2. **Relation classification**: predict the relation type between p_i (assume $p_i = u_j$) and u_i with a relation classifier. Similar to link prediction, the relation classifier leverages both non-structured and structured representations. Discourse representations of u_j and u_i are fed into an MLP to obtain a distribution over relation types. The relation type r_{ji} is taken with the largest probability.
3. **Structured representation encoding**: compute the structured representation of u_i with a structured representation encoder which encodes the predicted discourse structure. Specifically, the relation embedding of r_{ji}, the non-structured representation of u_i, and the structured representation of $p_i = u_j$, are fed into the encoder to derive the structured representation of u_i.

Afterwards, the model moves to the next EDU u_{i+1} and performs the above three steps iteratively until the end of the dialogue. In this manner, dependency prediction and discourse structure construction are performed jointly and alternately, and the discourse structure is built incrementally.

Discourse Representations

In our model, we use two categories of discourse representations: local representations and global representations. Local representations are non-structured and encode the local information of EDUs individually. And global representations encode the global information of the EDU sequence or the predicted discourse structure. These representations are taken as the input for link prediction and relation classification. In return, the predicted links and relation types are used to build structured global representations incrementally.

Local Representations For each EDU u_i, a bidirectional GRU (bi-GRU) encoder is applied on the word sequence, and the last hidden states in two directions are concatenated as the local representation of u_i, denoted as $\boldsymbol{h}_i$.

[4] We found that the proportion of EDUs with multiple incoming relations is quite limited (less than 6.4%) in the dataset we used. Nevertheless, our model can be easily extended to predict a more general DAG when necessary.

Non-structured Global Representations Non-structured global representations encode the EDU sequence in a dialogue. The local representations of the EDUs $\boldsymbol{h}_0, \boldsymbol{h}_1, \cdots, \boldsymbol{h}_n$ are taken as input to a GRU encoder and the hidden states are viewed as the *non-structured global representations* of the EDUs, denoted as $\boldsymbol{g}_0^{NS}, \boldsymbol{g}_1^{NS}, \cdots, \boldsymbol{g}_n^{NS}$.

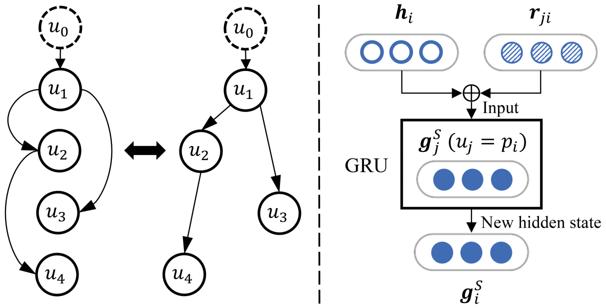


Figure 3: An example dependency tree (left) and the structured encoder (right), where $\boldsymbol{h}_i$ is the local representation of EDU u_i, $\boldsymbol{g}_i^S$ and $\boldsymbol{g}_j^S$ are structured representations, $\boldsymbol{r}_{ji}$ is the relation embedding, and $u_j = p_i$ is the parent of u_i.

Structured Global Representations The structured representations encode dependency links and the corresponding relation types to fully utilize the global information of the predicted structure. Note that there is exactly one path from the root to each EDU on the predicted dependency tree, and the path represents the development of the dialogue. We apply a structured encoder to these paths to obtain *structured global representations* (or *structured representations* briefly) of the EDUs. For instance, to obtain the structured representation of u_4 as shown in Figure 3, the structured encoder is applied to the path $u_0 \rightarrow u_1 \rightarrow u_2 \rightarrow u_4$, and the hidden state at u_4 is treated as its structured representation. The structured representations are computed incrementally. We compute the structured representation of u_i once its parent and the corresponding relation type are decided.

In addition, when predicting a dependency relation linking from u_j to u_i, it is beneficial to highlight previous utterances from the same speaker as that of u_i. Because it helps the model to better understand the development of the dialogue involving this speaker, which can improve the prediction of the dependency related to u_i. For instance, if we consider a dependency path like $\cdots \rightarrow u_k(a) \rightarrow \cdots \rightarrow u_j(b) \rightarrow u_i(a)$ where a, b are speaker identifiers, when predicting the dependency link between u_j and u_i, it is beneficial to highlight the previous dialogue history u_k from the same speaker as that of u_i, namely a. Therefore, we propose a *Speaker Highlighting Mechanism (SHM)*, with which we compute $|\mathcal{A}|$ different structured representations for each EDU such that each one highlights a specific speaker, where $\mathcal{A}$ is the set of all speakers in the dialogue. This is particularly effective for multi-party dialogues.

Let $\boldsymbol{g}_{i,a}^S$ denote the structured representation of u_i when highlighting speaker a, $p_i = u_j$ is the predicted parent of u_i, and a_i is the speaker of EDU u_i. We compute the structured representations as follows:

$$\boldsymbol{g}_{i,a}^S = \begin{cases} \mathbf{0} & i = 0 \\ \mathbf{GRU}_{hl}(\boldsymbol{g}_{j,a}^S, \boldsymbol{h}_i \oplus \boldsymbol{r}_{ji}) & a_i = a, i > 0 \\ \mathbf{GRU}_{gen}(\boldsymbol{g}_{j,a}^S, \boldsymbol{h}_i \oplus \boldsymbol{r}_{ji}) & a_i \neq a, i > 0 \end{cases} \quad (3)$$

where $\oplus$ denotes vector concatenation, $\mathbf{GRU}$ stands for the functions of a GRU cell, and $\boldsymbol{r}_{ji}$ denotes the embedding vector of relation type r_{ji}, and hl and gen are short for $highlighted$ and $general$ respectively.

In Eq. (3), $\boldsymbol{g}_{0,a}^S$ is set to a zero vector since the dummy root contains no real information. We compute $\boldsymbol{g}_{i,a}^S(i > 0)$ based on the structured representation of its parent $\boldsymbol{g}_{j,a}^S$ that also highlights speaker a, and we use two different GRU cells $\mathbf{GRU}_{hl}$ and $\mathbf{GRU}_{gen}$ to respect whether the current speaker a_i is highlighted or not. For the selected GRU cell, as shown in Figure 3, $\boldsymbol{g}_{j,a}^S$ is the previous hidden state, $\boldsymbol{h}_i \oplus \boldsymbol{r}_{ji}$ is the input at the current step, and the new hidden state becomes $\boldsymbol{g}_{i,a}^S(i > 0)$.

Link Prediction and Relation Classification

For each EDU u_i, the link predictor predicts its parent node p_i and the relation classifier categorizes the corresponding relation type r_{ji} if $p_i = u_j$. For each EDU $u_j(j < i)$ that precedes u_i in the dialogue, we concatenate the representations $\boldsymbol{h}_i, \boldsymbol{g}_i^{NS}, \boldsymbol{g}_j^{NS}, \boldsymbol{g}_{j,a_i}^S$ to obtain an input vector $\boldsymbol{H}_{i,j}$ for link prediction and relation classification:

$$\boldsymbol{H}_{i,j} = \boldsymbol{h}_i \oplus \boldsymbol{g}_i^{NS} \oplus \boldsymbol{g}_j^{NS} \oplus \boldsymbol{g}_{j,a_i}^S \quad (4)$$

For both u_i and u_j, their non-structured global representations $\boldsymbol{g}_i^{NS}$ and $\boldsymbol{g}_j^{NS}$ are included in the input. We also add $\boldsymbol{g}_{j,a_i}^S$ which is the structured representation of u_j when highlighting the speaker of u_i, namely a_i. And since the structured representation of u_i is unavailable at the current step, we add the local representation of u_i, namely $\boldsymbol{h}_i$ instead.

Taking $\boldsymbol{H}_{i,<i}$ $(\boldsymbol{H}_{i,<i} = \boldsymbol{H}_{i,0}, \cdots, \boldsymbol{H}_{i,i-1})$ as input, the link predictor estimates the probability that each $u_j(j < i)$ is the parent of u_i on the dependency tree. The relation classifier then predicts the relation type between u_j and u_i, if u_j is the predicted parent of u_i.

Link Prediction The link predictor first projects the input vectors $\boldsymbol{H}_{i,j}(j < i)$ to a hidden representation:

$$\boldsymbol{L}_{i,j}^{link} = \tanh(\boldsymbol{W}_{link} \cdot \boldsymbol{H}_{i,j} + \boldsymbol{b}_{link}) \quad (5)$$

where $\boldsymbol{W}_{link} \in \mathbb{R}^{d_l \times d_h}$ and $\boldsymbol{b}_{link} \in \mathbb{R}^{d_h}$ are parameters, d_l and d_h are dimensions of $\boldsymbol{L}_{i,j}^{link}$ and $\boldsymbol{H}_{i,j}$ respectively.

The predictor then computes the probability that u_j is the parent of u_i on the predicted dependency tree as follows:

$$o_{i,j}^{link} = \boldsymbol{U}_{link} \cdot \boldsymbol{L}_{i,j}^{link} + b'_{link} \quad (6)$$

$$P(p_i = u_j | \boldsymbol{H}_{i,<i}) = \frac{exp(o_{i,j}^{link})}{\sum_{k<i} exp(o_{i,k}^{link})} \quad (7)$$

where $\boldsymbol{U}_{link} \in \mathbb{R}^{1 \times d_l}$ and $b'_{link} \in \mathbb{R}$ are also parameters.

Hence, the predicted p_i is chosen as follows:

$$p_i = \underset{u_j : j<i}{\operatorname{argmax}} P(p_i = u_j | \boldsymbol{H}_{i,<i}) \quad (8)$$

Unlike the local classifiers in (Li et al. 2014; Afantenos et al. 2015; Perret et al. 2016), link prediction by $P(p_i = u_j|\boldsymbol{H}_{i,<i})$ depends on all candidate parents due to the $softmax$ normalization factor in Eq. (7). During training, the gradient of each candidate parent is also dependent on all candidate parents, from which it can utilize more information for training, while other methods consider each candidate parent individually.

Relation Classification Similar to link prediction, the relation classifier first projects the input vector $\boldsymbol{H}_{i,j}$ to a hidden representation, as follows:

$$\boldsymbol{L}_{i,j}^{rel} = \tanh(\boldsymbol{W}_{rel} \cdot \boldsymbol{H}_{i,j} + \boldsymbol{b}_{rel}) \tag{9}$$

where $\boldsymbol{W}_{rel} \in \mathbb{R}^{d_l \times d_h}, \boldsymbol{b}_{rel} \in \mathbb{R}^{d_l}$ are parameters and d_l equals to the dimension of $\boldsymbol{L}_{i,j}^{rel}$.

The classifier then predicts the relation type r_{ji} from the probability distribution over all types computed as follows:

$$P(r|\boldsymbol{H}_{i,j}) = softmax(\boldsymbol{U}_{rel} \cdot \boldsymbol{L}_{i,j}^{rel} + \boldsymbol{b}'_{rel}) \tag{10}$$

where $\boldsymbol{U}_{rel} \in \mathbb{R}^{K \times d_h}, \boldsymbol{b}'_{rel} \in \mathbb{R}^{K}$ are also parameters.

Loss Function

We adopt the negative log-likelihood of the training data as the loss function:

$$L_{link}(\Theta) = -\sum_{d \in \mathcal{D}} \sum_{i=1}^{n} \log P(p_i = p_i^*|\boldsymbol{H}_{i,<i}) \tag{11}$$

$$L_{rel}(\Theta) = -\sum_{d \in \mathcal{D}} \sum_{i=1}^{n} \log P(r_{ji} = r_{ji}^*|\boldsymbol{H}_{i,j}, u_j = p_i^*) \tag{12}$$

$$L_{all}(\Theta) = L_{link}(\Theta) + L_{rel}(\Theta) \tag{13}$$

where Θ is the set of parameters to be optimized, $\mathcal{D}$ is the training data, d is a dialogue in $\mathcal{D}$, p_i^* and r_{ji}^* are the golden parent and the corresponding golden relation type respectively.

Since the golden discourse structure is a dependency graph while our model predicts a dependency tree, to determine the golden parent p_i^* of each EDU u_i for training, we take the earliest EDU with a relation linking to u_i. And if u_i is not linked from any preceding unit, we set $p_i^* = u_0$.

In $L_{rel}(\Theta)$, we use the log-likelihood of the relation type between u_i and the golden parent p_i^* rather than the predicted p_i, because the link predictor may predict incorrect p_i such that the golden relation type between p_i and u_i can be unavailable.

Experiments

Data Preparation

We adopted the STAC Corpus (Asher et al. 2016)[5] which is a multi-party dialogue corpus collected from an online game. Its annotations follow Segmented Discourse Representation Theory (SDRT) (Asher and Lascarides 2003), where a discourse unit linked by a dependency relation may be an EDU, or a group of coherent discourse units named *Complex Discourse Unit (CDU)* (Asher et al. 2016).

Previous studies for discourse dependency parsing have suggested that detecting CDUs remains challenging, and they thus transformed SDRT structures to dependency structures by eliminating CDUs (Muller et al. 2012; Afantenos et al. 2015; Perret et al. 2016). Therefore, our task does not involve CDUs either. We adopted the strategy firstly proposed by (Muller et al. 2012) to replace the CDUs with their heads recursively, where the head of a CDU is the earliest discourse unit in it without incoming relations. This strategy was also adopted by (Afantenos et al. 2015; Perret et al. 2016). But we did not apply another strategy mentioned by (Perret et al. 2016) which clones the relation to link every discourse unit in a CDU, since we found that this strategy brings many redundant and inappropriate relations as shown in Figure 4, and therefore, it may mislead the parsing models.

After eliminating the CDUs, the dataset consists of 1,062 dialogues, 11,711 EDUs and 11,350 relations in the training data; and 111 dialogues, 1,156 EDUs and 1,126 relations in the test data. We retained 10% of the training dialogues for validation. Similar to prior studies, for each dialogue, we manually added a relation from the dummy root to each EDU without an incoming relation, with a special relation type *ROOT*. Moreover, we discarded the dialogue act annotations on EDUs in the original dataset as they are irrelevant to our problem.

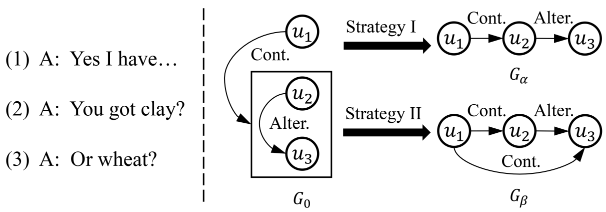


Figure 4: An example of eliminating a CDU which consists of u_2 and u_3 from the original SDRT structure G_0. "Cont." is short for "Continuation" and "Alter." for "Alternation". There are two strategies: *Strategy I* links u_1 to the head of the CDU ($u_1 \rightarrow u_2$), resulting in G_α; and *Strategy II* duplicates the link to every unit of the CDU ($u_1 \rightarrow u_2; u_1 \rightarrow u_3$), resulting in G_β. The relation $u_1 \rightarrow u_3$ in G_β is inappropriate, since u_3 is not a direct continuation of u_1. We took *Strategy I* as it appears to be more reasonable in most cases.

Baselines

We adopted the following baselines for comparison:

- **MST** (Afantenos et al. 2015): A two-stage approach that adopts Maximum Spanning Trees (MST) to construct the discourse structure. MST builds a dependency tree using the probabilities from a dependency relation classifier which uses local information only.

[5] https://www.irit.fr/STAC/corpus.html. We used the version released on March 21, 2018.

- **ILP** (Perret et al. 2016) : A variant of MST which replaces the MST algorithm with Integer Linear Programming (ILP) to construct the discourse structure.
- **Deep+MST**: a variant of MST which uses discourse representations from GRU encoders, instead of hand-crafted features or external parsers. These representations are similar to those of our deep sequential model, but they only include non-structured representations.
- **Deep+ILP**: A variant of ILP with the same modification as from MST to Deep+MST.
- **Deep+Greedy**: It is similar to Deep+MST and Deep+ILP, but this model adopts a greedy decoding algorithm which directly selects a parent for each EDU from previous EDUs with the largest probability.

We evaluated *MST* and *ILP* using the open source code from (Afantenos et al. 2015; Perret et al. 2016)[6]. As for the deep baseline models, since the structured representations are unavailable, we replaced $\boldsymbol{g}^S_{j,a_i}$ in $\boldsymbol{H}_{i,j}$ with $\boldsymbol{h}_j$ instead, and thus the input for dependency prediction becomes:

$$\boldsymbol{H}'_{i,j} = \boldsymbol{h}_i \oplus \boldsymbol{g}_i^{NS} \oplus \boldsymbol{h}_j \oplus \boldsymbol{g}_j^{NS} \tag{14}$$

where we concatenate the non-structured representations of EDU u_i and u_j together. Moreover, to compare the models fairly, the dimensions of the input vector in all the deep baseline models and our sequential model are kept the same.

Implementation Details

The word vectors are initialized with 100-dimensional Glove vectors (Pennington, Socher, and Manning 2014) and are fine-tuned during training. The dimensions of the relation embeddings an the discourse representations are set to 100 and 256 respectively. And the dimensions of the hidden representations in link prediction and relation classification are set to 512. Dropout is adopted before the input of each GRU cell, with a probability of 0.5. We use Stochastic Gradient Descent (SGD) to train the model, with the mini-batch size set to 4. The initial learning rate is set to 0.1 and it decays at a constant rate of 0.98 after each epoch.

Moreover, we experimented with two settings of our deep sequential model. One is a shared version where the link predictor and the relation classifier share the same input vector $\boldsymbol{H}_{i,j}$. The other is a non-shared version where the two input vector $\boldsymbol{H}_{i,j}$ in Eq. (5) and that in Eq. (9) are from networks with different parameters respectively. We finally took the later one and also applied it to deep baseline models.

Results

We adopted micro-averaged F_1 score as the evaluation metric. Results for different models are shown in Table 1, where "Link" denotes link prediction while "Link & Rel" stands for that a correct prediction must predict dependency link and relation type correctly at the same time.

Our deep sequential model outperforms all the baselines significantly (bootstrap test, $p < 0.05$), demonstrating the benefit of predicting dependency relations and constructing the discourse structure jointly and alternately. Besides, we observed that the performance drop when link prediction and relation classification share the same discourse representations (*Deep Sequential (Shared)*). This is probably because that it is hard to train the discourse encoders to simultaneously capture the information needed by both link prediction and relation classification.

Model	Link	Link & Rel
MST	68.8	50.4
ILP	68.6	52.1
Deep+MST	69.6	52.1
Deep+ILP	69.0	53.1
Deep+Greedy	69.3	51.9
Deep Sequential (shared)	72.1	54.7
Deep Sequential	**73.2**	**55.7**

Table 1: F_1 scores (%) for different models. *Link* means link prediction; and *Link & Rel* means that a correct prediction must predict dependency link and relation type correctly at the same time.

Moreover, compared to *MST* and *ILP* that rely on hand-crafted features and external parsers, the deep baseline models *Deep+MST* and *Deep+ILP* achieve higher F_1 scores. This demonstrates that discourse representations from hierarchical GRU encoders are more effective than traditional features. *Deep+Greedy* has a lower F_1 score on *Link & Rel* compared to *Deep+MST* and *Deep+ILP*, indicating that more sophisticated decoding algorithms help construct better structures. Interestingly, our deep sequential model, which does not have any complex decoding algorithm, still outperforms those baselines. This further validates the effectiveness of our sequential model.

Effectiveness of the Structured Representations

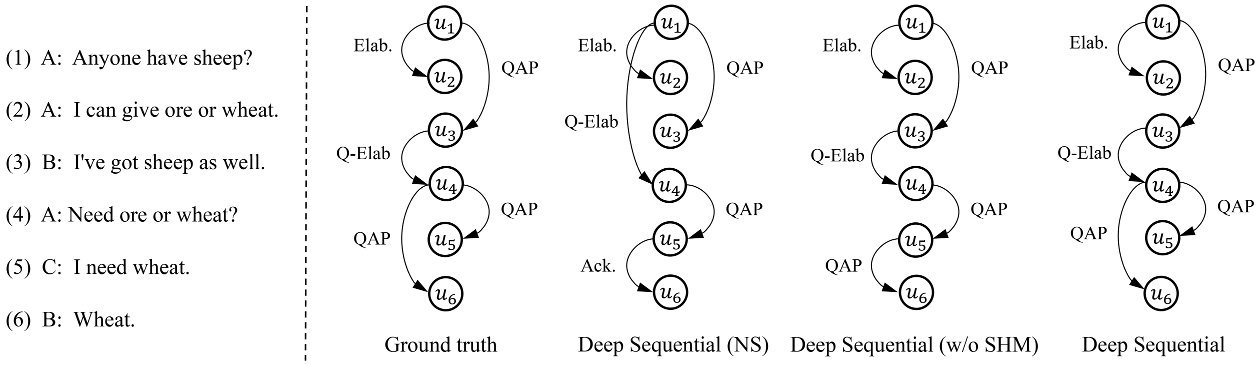
To evaluate the effectiveness of the structured representations, we devised the following three variants of our deep sequential model for comparison:

- **Deep Sequential (NS)**: We removed the structured representations from our original deep sequential model. Similar to that of other deep baselines, the input to the link predictor and relation type classifier has only non-structured representations, as defined by Eq. (14).
- **Deep Sequential (Random)**: In this variant, both non-structured and structured representations are used, but the structured representations encode a random structure. For each EDU, we randomly choose a parent from the preceding EDUs and a random relation type, to obtain its structured representation.
- **Deep Sequential (w/o SHM)**: We disabled the speaker highlighting mechanism in the full model to evaluate the effectiveness of this mechanism.

Results in Table 2 reveal the following observations:

1. Our full model (*Deep Sequential*) outperforms *Deep Sequential (NS)* and *Deep Sequential (Random)*, indicating

[6] https://github.com/irit-melodi/irit-stac



(1) A: Anyone have sheep?

(2) A: I can give ore or wheat.

(3) B: I've got sheep as well.

(4) A: Need ore or wheat?

(5) C: I need wheat.

(6) B: Wheat.

Figure 5: A dialogue example from three speakers, along with the golden discourse structure and discourse structures predicted by various models. "Elab." is short for "Elaboration", "QAP" for "Question-Answer-Pair", "Q-Elab" for "Question-Elaboration", and "Ack." for "Acknowledgement". u_i in the graphs corresponds to the i-th utterance in the left panel.

Model	Link	Link & Rel
Deep+Greedy	69.3	51.9
Deep Sequential (NS)	71.0	53.7
Deep Sequential (Random)	71.8	53.7
Deep Sequential (w/o SHM)	71.7	54.5
Deep Sequential	**73.2**	**55.7**

Table 2: F_1 scores (%) for different models.

that the structured representations which encode the predicted discourse structure are crucial for dependency relation prediction.

2. When the speaker highlighting mechanism is disabled (*Deep Sequential (w/o SHM)*), there is a remarkable drop on performance, which demonstrates that the speaker highlighting mechanism can improve the prediction of dependency relations.

3. A random structure can help link prediction slightly, as can be seen from the comparative results between *Deep Sequential (Random)* and *Deep Sequential (NS)* (71.8 vs. 71.0). However, *Deep Sequential* is much better than *Deep Sequential (Random)*, indicating that structured representations can effectively encode valuable information from the predicted discourse structure.

We also noticed that *Deep Sequential (NS)* still has an improvement over *Deep+Greedy*, even without structured representations. It is because the probabilities of dependency links are dependent on all candidate parents in the sequential models due to the $softmax$ normalization, whereas the baseline models predict each dependency link individually. Thereby, the global information from other candidate links benefits the training of sequential models.

Case Study

We provide an example to show how structured information helps the model to better understand the development of the dialogue, which is important for dependency prediction.

As shown in Figure 5, *Deep Sequential (NS)* incorrectly predicts the parent of u_4 as u_1 while the ground truth is u_3, yet both *Deep Sequential (w/o SHM)* and *Deep Sequential* make correct predictions. The previously predicted dependency relation $u_1 \rightarrow u_3$ with type *QAP*, is encoded by structured representations. It thus helps the model to understand that the question in u_1 has been answered by u_3, and therefore, it is more likely that u_4 responds to u_3, rather than elaborates the original question u_1 which has already been responded by others.

Moreover, both *Deep Sequential (NS)* and *Deep Sequential (w/o SHM)* incorrectly predict the parent of u_6 while *Deep Sequential* makes a correct prediction. Thanks to the speaker highlighting mechanism, when predicting the parent of u_6, the model highlights the previous EDU u_3 from the same speaker as that of u_6 (i.e. speaker B) in the structured representations. Thereby, in order to predict a dependency relation $u_4 \rightarrow u_6$, the model tends to utilize the information that it is u_4 which responds to the previous EDU u_3 by the speaker of u_6.

Conclusion and Future Work

In this paper, we propose a deep sequential model for discourse parsing on multi-party dialogues. The model predicts dependency relations and builds the discourse structure jointly and alternately. It decides the dependency links between EDUs and the corresponding relation types sequentially, utilizing the structured representation of each EDU encoded with a structured encoder, and in return, the predicted dependency relations are used to construct the discourse structure incrementally. Experiments show that our sequential model outperforms all the state-of-the-art baselines significantly and the structured representations can effectively improve dependency prediction.

We not only propose an approach to parse discourse structures, but also an approach to utilize them via a structured encoder. We have further demonstrated the benefit of discourse structures. As future work, our method can be enhanced and applied to improve approaches for other NLP tasks of multi-party dialogues.

Acknowledgments

This work was jointly supported by the National Science Foundation of China (Grant No.61876096/61332007), and the National Key R&D Program of China (Grant No. 2018YFC0830200). We would like to thank Prof. Xiaoyan Zhu for her generous support.

References

Afantenos, S.; Kow, E.; Asher, N.; and Perret, J. 2015. Discourse parsing for multi-party chat dialogues. In *ACL*, 928–937.

Asher, N., and Lascarides, A. 2003. *Logics of conversation*. Cambridge University Press.

Asher, N.; Hunter, J.; Morey, M.; Benamara, F.; and Afantenos, S. D. 2016. Discourse structure and dialogue acts in multiparty dialogue: the stac corpus. In *LREC*, 2721–2727.

Bhatia, P.; Ji, Y.; and Eisenstein, J. 2015. Better document-level sentiment analysis from rst discourse parsing. In *EMNLP*, 2212–2218.

Braud, C.; Coavoux, M.; and Søgaard, A. 2017. Cross-lingual rst discourse parsing. *arXiv preprint arXiv:1701.02946*.

Cambria, E.; Schuller, B.; Xia, Y.; and Havasi, C. 2013. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems* 28(2):15–21.

Chen, Z.; Yang, R.; Zhao, Z.; Cai, D.; and He, X. 2018. Dialogue act recognition via crf-attentive structured network. In *SIGIR*, 225–234. ACM.

Cho, K.; Gulcehre, B. v. M. C.; Bahdanau, D.; Schwenk, F. B. H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 1724–1734.

Feng, V. W., and Hirst, G. 2014. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 511–521.

Jia, Y.; Feng, Y.; Ye, Y.; Lv, C.; Shi, C.; and Zhao, D. 2018a. Improved discourse parsing with two-step neural transition-based model. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 17(2):11.

Jia, Y.; Ye, Y.; Feng, Y.; Lai, Y.; Yan, R.; and Zhao, D. 2018b. Modeling discourse cohesion for discourse parsing via memory network. In *ACL*, volume 2, 438–443.

Joty, S.; Carenini, G.; and Ng, R. T. 2015. Codra: A novel discriminative framework for rhetorical analysis. *Computational Linguistics* 41(3):385–435.

Kumar, H.; Agarwal, A.; Dasgupta, R.; Joshi, S.; and Kumar, A. 2018. Dialogue act sequence labeling using hierarchical encoder with crf. In *AAAI*, 3440–3447.

Li, S.; Wang, L.; Cao, Z.; and Li, W. 2014. Text-level discourse dependency parsing. In *ACL*, volume 1, 25–35.

Li, Q.; Li, T.; and Chang, B. 2016. Discourse parsing with attention-based hierarchical neural networks. In *EMNLP*, 362–371.

Li, J.; Li, R.; and Hovy, E. 2014. Recursive deep models for discourse parsing. In *EMNLP*, 2061–2069.

Liu, Y., and Lapata, M. 2017. Learning contextually informed representations for linear-time discourse parsing. In *EMNLP*, 1289–1298.

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

McDonald, R.; Pereira, F.; Ribarov, K.; and Hajič, J. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 523–530. Association for Computational Linguistics.

Morey, M.; Muller, P.; and Asher, N. 2018. A dependency perspective on rst discourse parsing and evaluation. *Computational Linguistics* 198–235.

Muller, P.; Afantenos, S.; Denis, P.; and Asher, N. 2012. Constrained decoding for text-level discourse parsing. In *COLING*, 1883–1900.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, 1532–1543.

Perret, J.; Afantenos, S.; Asher, N.; and Morey, M. 2016. Integer linear programming for discourse parsing. In *NAACL*, 1883–1900.

Prasad, R.; Miltsakaki, E.; Dinesh, N.; Lee, A.; Joshi, A.; Robaldo, L.; and Webber, B. L. 2007. The penn discourse treebank 2.0 annotation manual.

Seo, J.; Croft, W. B.; and Smith, D. A. 2009. Online community search using thread structure. In *Proceedings of the 18th ACM conference on Information and knowledge management*, 1907–1910. ACM.

Takanobu, R.; Huang, M.; Zhao, Z.; Li, F.-L.; Chen, H.; Zhu, X.; and Nie, L. 2018. A weakly supervised method for topic segmentation and labeling in goal-oriented dialogues via reinforcement learning. In *IJCAI*, 4403–4410.

Verberne, S.; Boves, L.; Oostdijk, N.; and Coppen, P.-A. 2007. Evaluating discourse-based answer extraction for why-question answering. In *SIGIR*, 735–736. ACM.

Wang, Y.; Li, S.; and Wang, H. 2017. A two-stage parsing method for text-level discourse analysis. In *ACL*, volume 2, 184–188.

Yang, A., and Li, S. 2018. Scidtb: Discourse dependency treebank for scientific abstracts. *arXiv preprint arXiv:1806.03653*.

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