

A Label Dependence-aware Sequence Generation Model for Multi-level Implicit Discourse Relation Recognition

Changxing Wu¹, Liuwen Cao¹, Yubin Ge², Yang Liu³, Min Zhang⁴, Jinsong Su^{5*}

¹ School of Software, East China Jiaotong University, Nanchang, China

² University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

³ Tsinghua University, Beijing, China ⁴ Soochow University, Soochow, China ⁵ Xiamen University, Xiamen, China

wuchangxing@ecjtu.edu.cn, caoliuwenc@163.com, yubinge2@illinois.edu,
 liuyang2011@tsinghua.edu.cn, minzhang@suda.edu.cn, jssu@xmu.edu.cn

Abstract

Implicit discourse relation recognition (IDRR) is a challenging but crucial task in discourse analysis. Most existing methods train multiple models to predict multi-level labels independently, while ignoring the dependence between hierarchically structured labels. In this paper, we consider multi-level IDRR as a conditional label sequence generation task and propose a **Label Dependence-aware Sequence Generation Model** (LDGSM) for it. Specifically, we first design a label attentive encoder to learn the global representation of an input instance and its level-specific contexts, where the label dependence is integrated to obtain better label embeddings. Then, we employ a label sequence decoder to output the predicted labels in a top-down manner, where the predicted higher-level labels are directly used to guide the label prediction at the current level. We further develop a mutual learning enhanced training method to exploit the label dependence in a bottom-up direction, which is captured by an auxiliary decoder introduced during training. Experimental results on the PDTB dataset show that our model achieves the state-of-the-art performance on multi-level IDRR. We release our code at <https://github.com/nlpersECJTU/LDGSM>.

Introduction

As a crucial task in discourse analysis, implicit discourse relation recognition (IDRR) aims to identify semantic relations (e.g., *Contingency*) between two arguments (sentences or clauses). Figure 1 shows an instance from PDTB (Prasad et al. 2008). It consists of two arguments (arg_1 and arg_2) and is annotated with three hierarchical labels¹, where the second-level/child label *Cause* further refines the semantic of the top-level/parent label *Contingency*, and so on. During annotation, an implicit connective *because* is first inserted to benefit the label annotation, and can be considered as the most fine-grained label. Compared with explicit discourse relation recognition, IDRR is more challenging because of the absence of explicit connectives. Due to its wide applications in many natural language processing (NLP) tasks,

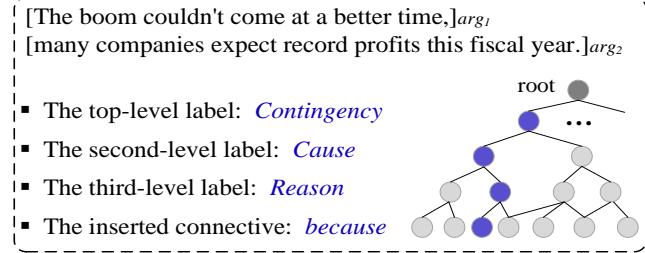


Figure 1: An implicit PDTB instance with annotated multi-level labels and the inserted connective. Labels defined in PDTB are organized in a three-layer hierarchical structure, and the inserted connectives can be considered as the fourth-level (most fine-grained) labels. For an instance, its multi-level labels and the inserted connective can be considered as a label sequence, e.g., *Contingency*, *Cause*, *Reason*, *because* (the path of blue nodes) for the above instance.

such as summarization (Cohan et al. 2018) and event relation extraction (Tang et al. 2021), IDRR has become one of hot research topics recently.

With the rapid development of deep learning, previous IDRR models relying on human-designed features have evolved to neural network based models (Zhang et al. 2015; Liu and Li 2016; Lei et al. 2017; Xu et al. 2019; Liu et al. 2020; Zhang et al. 2021). Despite their success, these studies usually train multiple models to predict multi-level labels independently, while ignoring the dependence between these hierarchically structured labels. To address this issue, Nguyen et al. (2019) defined three loss terms to leverage the dependence between labels and connectives. Though achieving promising results, they only exploit the label-connective dependence to guide the model training. Wu et al. (2020) introduced a hierarchical multi-task neural network stacked with a CRF layer to infer the sequence of multi-level labels. However, there still exist two drawbacks: 1) They stack several feature layers together, and directly use features from different layers for label predictions at different levels, which may be insufficient to fully exploit the label dependence for feature extraction. 2) During decoding, the CRF layer has the limitation in processing the label dependence because it

*Corresponding author

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

¹It is noteworthy that the organization of discourse relation labels is predominantly hierarchical, which holds in the corpora of different languages, for example, CDTB (Li et al. 2014).

only uses transition matrices to encourage valid label paths and discourage other paths (Collobert et al. 2011). In other words, the label predictions at higher levels are not directly used to guide the label prediction at the current level. Therefore it can be said that the potential of the dependence between hierarchically structured labels has not been fully exploited for multi-level IDRR.

In this paper, we consider multi-level IDRR as a conditional label sequence generation task and propose a **Label Dependence-aware Sequence Generation Model** (LDSGM). Overall, our model consists of a label attentive encoder and a label sequence decoder, both of which can effectively leverage the dependence between labels. In the encoder, based on the hierarchical structure of labels, we introduce graph convolutional network (Kipf and Welling 2017) to learn better label embeddings, which are used as the query of attention mechanisms to extract level-specific contexts. Then the decoder sequentially outputs the predicted labels level by level in a top-down manner. In this way, the easily-predicted higher-level labels can be directly used to guide the label prediction at the current level.

We further develop a mutual learning enhanced training method to leverage the label dependence. Concretely, we introduce an auxiliary decoder that sequentially conducts label predictions in a bottom-up manner. The intuition behind is that annotators usually first insert suitable connectives to facilitate the higher-level label annotations (Prasad et al. 2008; Li et al. 2014). Our decoder and the auxiliary one are able to capture the label dependence in two different directions and thus are complement to each other. Therefore, we transfer the knowledge of the auxiliary decoder to enhance ours via mutual learning (Zhang et al. 2018b). Please note that the auxiliary decoder is not involved during inference.

The main contributions of our work are three-fold:

- We consider multi-level IDRR as a label sequence generation task. To our knowledge, our work is the first attempt to deal with this task in the way of sequence generation.
- We propose a label dependence-aware sequence generation model. Both its encoder and decoder, combined with the mutual learning enhanced training method, can fully leverage the label dependence for multi-level IDRR.
- Experimental results and in-depth analysis show that our model achieves the state-of-the-art performance on the PDTB dataset, and more consistent predictions on multi-level labels.

Our Model

Given M hierarchical levels of defined labels $\mathbb{C} = (C^1, \dots, C^m, \dots, C^M)$, where C^m is the set of labels in the m -th hierarchical level, and an instance $\mathbf{x} = (arg_1, arg_2)$ as input, our LDSGM model aims to generate a label sequence $\mathbf{y} = y_1, \dots, y_m, \dots, y_M$, where $y_m \in C^m$. As shown in Figure 2, our model mainly consists of a label attentive encoder and a label sequence decoder. In the following sections, we will detail these two components and the mutual learning enhanced training method for our model.

Label Attentive Encoder

Our encoder comprises several stacked Transformer layers, a graph convolutional network (GCN) and level-specific label attention mechanisms. Specifically, we employ the Transformer layers to learn the local and global representations of an input instance, the GCN to obtain better label embeddings by integrating the dependence between hierarchically-structured labels, and finally the label attention mechanisms to extract level-specific contexts from the local representations. Afterwards, the learned global representation and level-specific contexts are used as inputs of our decoder for label sequence generation.

Given an instance $\mathbf{x} = (arg_1, arg_2)$, we first stack K Transformer layers (Vaswani et al. 2017) to learn its word-level semantic representations as follows:

$$v_{cls}, v_1, \dots, v_i, \dots, v_{N_w} = \text{Transformers}(arg_1, arg_2), \quad (1)$$

where the instance is formulated as a sequence with special tokens: [CLS]+ arg_1 +[SEP]+ arg_2 +[SEP], v_{cls} is the representation of [CLS], and $\{v_i\}_{i=1}^{N_w}$ are representations of other tokens. The key part of a Transformer layer is the multi-head self-attention mechanism. For IDRR, by concatenating two arguments as input, this mechanism can directly capture the semantic interactions between words within not only the same argument but also different arguments. During the subsequent predictions, v_{cls} is considered as the global representation of the instance and thus involved for the label predictions at each level, while $\{v_i\}_{i=1}^{N_w}$ are local representations from which we extract level-specific contexts for predictions at different levels.

Besides, we adopt a GCN to learn better label embeddings based on the hierarchical structure of labels. GCN is a multi-layer graph-based neural network that induces node embeddings based on properties of their neighborhoods (Kipf and Welling 2017). In this work, we directly consider the hierarchical structure of labels as an undirected graph, where each label corresponds to a graph node. The corresponding adjacent matrix $A \in \mathbb{R}^{N_C \times N_C}$ is defined as follows:

$$A_{jk} = \begin{cases} 1 & \text{if } j = k \\ 1 & \text{if } \text{Parent}(j) = k \text{ or } \text{Parent}(k) = j, \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $\text{Parent}(j) = k$ means that the label corresponding to node j is the subclass of that to node k , and N_C is total number of nodes (labels). Using GCN, given the initial representation of node j as e_j^0 , we produce the output e_j^l of node j at the l -th layer ($l \in [1, L]$) as

$$e_j^l = \sigma \left(\sum_{k=1}^{N_C} A_{jk} W^l e_k^{l-1} + b^l \right), \quad (3)$$

where W^l and b^l are the parameters, $\sigma(*)$ is a nonlinear function. Finally, we obtain the label embeddings as the outputs $\{e_j^L\}_{j=1}^{N_C}$ of the L -th layer. For simplicity, we denote all label embeddings at the m -th level as $E^m \in \mathbb{R}^{d_e \times |C^m|}$, with d_e as the dimension of label embedding.

Our GCN is a simple network with only parameters $W^l \in \mathbb{R}^{d_e \times d_e}$ and $b^l \in \mathbb{R}^{d_e}$ at the l -th layer. We select this simple GCN because discourse label graphs are usually small,

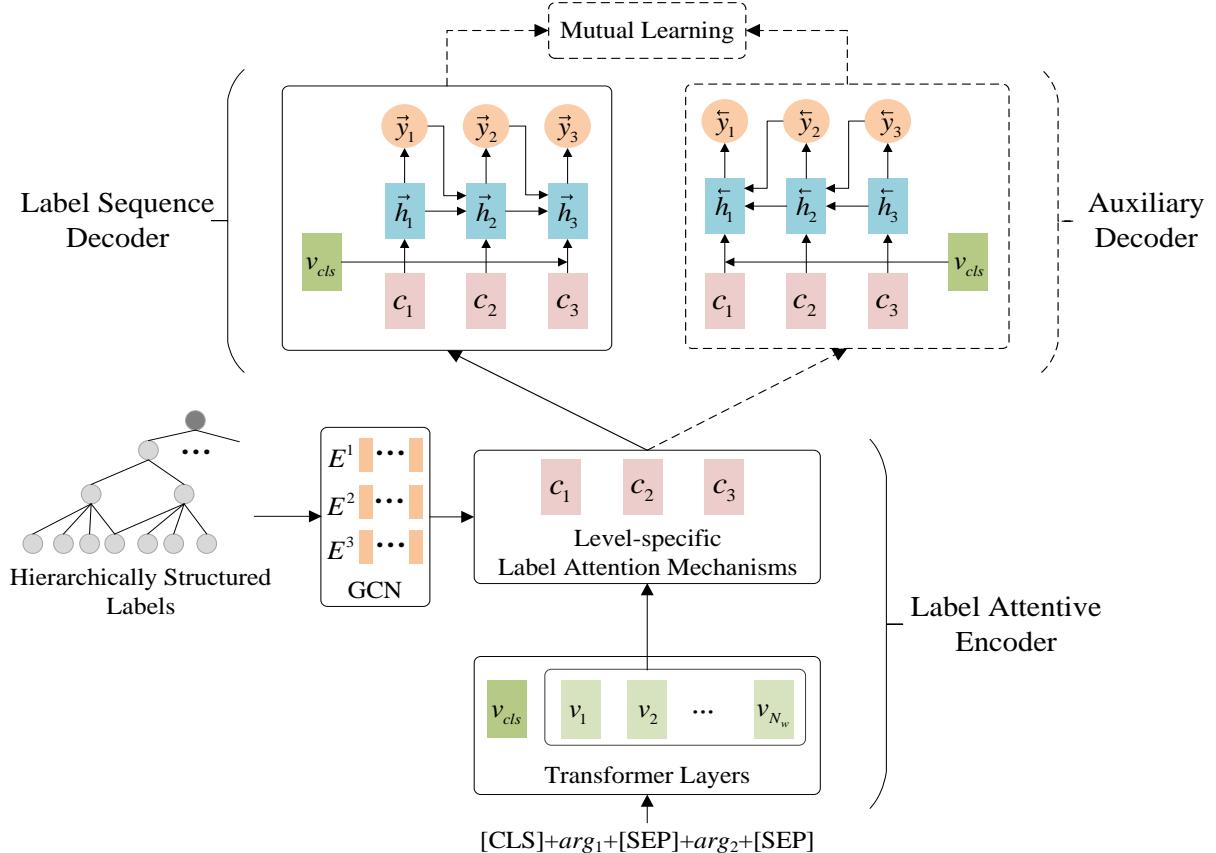


Figure 2: The architecture of our LDSGM model. The dotted parts are only included at the training phase.

for example, only 117 nodes and 262 edges in our experiments. It is also noteworthy that the label hierarchy is a graph (not tree) structure because many-to-many mappings exist between the second-level labels and connectives (some connectives are ambiguous).

In our encoder, one of the important steps is to capture the associations between input arguments and labels at each level. To this end, we design level-specific label attention mechanisms to learn contexts for label predictions. Specifically, given the label embeddings E^m and the learned local representations $V = \{v_1, \dots, v_i, \dots, v_{N_w}\} \in \mathbb{R}^{d_w \times N_w}$, we adopt the attention mechanism at the m -th level to extract the level-specific context c_m in the following way:

$$c_m = V\alpha, \\ \alpha = \text{softmax}(\text{max-pooling}(V^\top W^m E^m)), \quad (4)$$

where $W^m \in \mathbb{R}^{d_w \times d_e}$ is a parameter matrix, $\alpha \in \mathbb{R}^{N_w}$ is the calculated weight vector, $\text{max-pooling}(\ast)$ is the column-wise max-pooling operator, and \top denotes the matrix transpose operation. The i -th element in α indicates the association degree between the i -th local representation and all label embeddings at the m -th level. Note that all label embeddings at a level are used to query its level-specific context. Finally, we use the learned contexts $\{c_m\}_{m=1}^M$ for the label predictions at corresponding levels, respectively.

Overall, our encoder has several advantages: 1) The dependence between hierarchically structured labels is fully leveraged for better label embeddings. 2) By using the label attention mechanisms, our encoder extracts label-aware level-specific contexts for the predictions at different levels.

Label Sequence Decoder

Our decoder is an RNN-based one that sequentially generates the predicted labels in a top-down manner, that is, the top-level label, the second-level label, and so on. By doing so, the easily-predicted higher-level labels can be used to guide the label prediction at the current level. We choose Gate Recurrent Unit (GRU) (Cho et al. 2014) to construct our decoder because of its wide usage in text generation and the short length of label sequences. Please note that Transformer can also be used as our decoder.

Formally, given three kinds of inputs, including the global representation v_{cls} , the extracted level-specific context c_m , and the previously-predicted label distribution \vec{y}_{m-1} , our decoder produces the distribution \vec{y}_m at the m -th level as follows:

$$\vec{y}_m = \text{softmax}(W_o \vec{h}_m + b_o), \\ \vec{h}_m = \text{GRU}(\vec{h}_{m-1}, [v_{cls}; c_m; g(\vec{y}_{m-1})]), \quad (5)$$

where \vec{h}_m and \vec{h}_{m-1} are the decoder hidden states, $[;]$ is

the concatenation operation, and W_o and b_o are the trainable parameters. As for $g(\vec{y}_{m-1})$, only using the label with the highest probability under the distribution \vec{y}_{m-1} may cause error propagation. To mitigate this issue, we follow Yang et al. (2018) to fully exploit information in \vec{y}_{m-1} by defining $g(\vec{y}_{m-1})$ as $E^{m-1}\vec{y}_{m-1}$, where E^{m-1} represents all label embeddings at the level $m-1$. Our decoder exploits high-level predictions in a soft constraint manner. That is to say, it does not constrain the current predicted label to be a subclass of the previously-predicted label.

The CRF-based IDRR model (Wu et al. 2020) first computes label predictions at all levels simultaneously, and then infers the globally optimized label sequence. It does not leverage the predicted higher-level labels when computing the label prediction at current level. By contrast, our decoder exploits the predicted label distributions at higher levels for the current prediction, as described in Equation 5.

Mutual Learning Enhanced Training

As described above, our decoder generates labels in a top-down manner, which can only exploit the dependence from the predicted higher-level labels, leaving that from lower levels unexploited. To address this issue, we develop a mutual learning enhanced training method, where the label dependence in the bottom-up direction can be utilized to improve the model training.

As shown in the dotted parts of Figure 2, we introduce an auxiliary decoder that possesses the same architecture as the decoder but sequentially generates predicted label distributions $\vec{y}_M, \dots, \vec{y}_m, \dots, \vec{y}_1$ in a bottom-up manner². In this way, the predicted lower-level labels can be used to help the label prediction at the current level. The basic intuition behind our auxiliary decoder is that annotators usually insert a suitable connective to help the higher-level label annotations of a given instance. These inserted connectives can be considered as the most fine-grained labels, and have been exploited to boost IDRR in previous studies (Qin et al. 2017; Bai and Zhao 2018; Nguyen et al. 2019). Apparently, ours and the auxiliary decoder are able to capture the label dependence in two different directions and thus are complement to each other. Therefore, during the model training, we transfer the knowledge of the auxiliary decoder to enhance ours via mutual learning (Zhang et al. 2018b). Please note that our auxiliary decoder is not involved during inference.

To facilitate the understanding of the training procedure, we describe it in Algorithm 1. The most notable characteristic is that ours and the auxiliary decoder can boost each other by iteratively transferring knowledge between them during training. To do this, in addition to the conventional cross-entropy based losses, we introduce two additional losses to minimize the divergence between the predicted label distributions of these two decoders. Concretely, suppose θ_e , θ_d and θ_{ad} to be the parameter sets of the encoder, the decoder, and the auxiliary decoder, we define the following objective functions over the training data D to update these parame-

²The auxiliary decoder also uses the learned global representation v_{cls} and the level-specific contexts $\{c_m\}_{m=1}^M$.

Algorithm 1: The training procedure

```

1: Input: Training set  $D$ , validation set  $D'$ 
2: repeat
3:   repeat
4:     Load a batch size of instances  $B \in D$ 
5:     Generate predicted label distributions  $\vec{y}_1, \dots, \vec{y}_m, \dots, \vec{y}_M$  using the decoder for each instance in  $B$ 
6:     Generate predicted label distributions  $\vec{y}_M, \dots, \vec{y}_m, \dots, \vec{y}_1$  using the auxiliary decoder for each instance in  $B$ 
7:     Update  $\theta_e, \theta_d$  by minimizing  $L(B; \theta_e, \theta_d)$ 
8:     Update  $\theta_{ad}$  by minimizing  $L(B; \theta_{ad})$ 
9:     Save the best model according to the average performance at all levels on  $D'$ 
10:    until no more batches
11:   until convergence

```

ters, respectively:

$$L(D; \theta_e, \theta_d) = \sum_{(\mathbf{x}, \mathbf{y}) \in D} \sum_{m=1}^M \{-\mathbb{E}_{y_m} [\log \vec{y}_m] \\ + \lambda * \text{KL}(\vec{y}_m || \vec{y}_m)\}, \quad (6)$$

$$L(D; \theta_{ad}) = \sum_{(\mathbf{x}, \mathbf{y}) \in D} \sum_{m=1}^M \{-\mathbb{E}_{y_m} [\log \vec{y}_m] \\ + \lambda * \text{KL}(\vec{y}_m || \vec{y}_m)\},$$

where y_m is the one-hot encoding of the ground-truth label $y_m \in \mathbf{y}$, \vec{y}_m and \vec{y}_m are the predicted label distributions at the m -th level by ours and the auxiliary decoder respectively, $\mathbb{E}_{y_m}[\cdot]$ is the expected value with respect to y_m , $\text{KL}(\cdot || \cdot)$ is the Kullback-Leibler(KL) divergence function, and λ is the coefficient used to control the impacts of different loss items. We repeat the above knowledge transfer process, until both loss functions converge. By doing so, the label dependence in both top-down and bottom-up directions can be captured to benefit the decoder during the model training.

Experiments

To investigate the effectiveness of our model, we conduct experiments on the PDTB 2.0 corpus (Prasad et al. 2008).

Dataset

Since only part of PDTB instances are annotated with the third-level labels, we take the top-level and second-level labels into consideration, and regard the inserted connectives as the third-level labels. There are four top-level labels are considered, including *Temporal (Temp)*, *Contingency (Cont)*, *Comparison (Comp)* and *Expansion (Expa)*. Further, there exist 16 second-level labels, five of which with few training instances and no validation and test instance are removed. Therefore, we conduct an 11-way classification on the second-level labels. For the connective classification, we consider all 102 connectives defined in PDTB. Following previous studies (Bai and Zhao 2018; Wu et al. 2020; Liu et al. 2020), we split the PDTB corpus into three parts: the training set with 12,775 instances (Section 2-20), the validation set with 1,183 instances (Section 0-1), and the test set with 1,046 instances (Section 21-22).

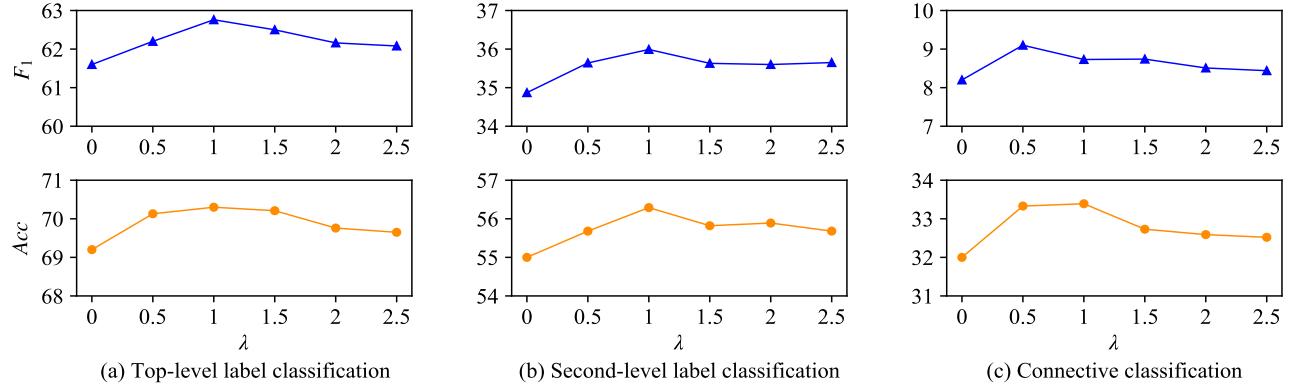


Figure 3: Effects of λ on the validation set.

Settings

All hyper-parameters are determined according to the best average model performance at three levels on the validation set. We use the pre-trained RoBERTa-base³ (Liu et al. 2019), which has 12 layers with 768 hidden units and 12 attention heads per layer, as our Transformer layers. We set the layer number of GCN as 2 since more layers do not bring the performance improvement. The graph node embeddings with the dimension 100 are randomly initialized according to the uniform distribution $U[-0.01, 0.01]$. The GRU hidden states in decoders are 768 dimensions. All other parameters are initialized with the default values in PyTorch. To avoid overfitting, we employ the *dropout* strategy (Srivastava et al. 2014) with the rate 0.2 over the outputs of RoBERTa-base. Finally, we adopt Adam (Kingma and Ba 2015) with the learning rate of $e-5$ and the batch size of 32 to update the model parameters for 15 epochs. To alleviate instability in the model training, we conduct all experiments for five times with different random seeds and report average results. Due to the skewed distribution of test dataset, we use both the macro-averaged F_1 score and accuracy (Acc) to evaluate the model.

Baselines

We compare our LDSGM model with the following competitive baselines:

- IDRR-C&E (Dai and Huang 2019): a neural IDRR model using a regularization approach to leverage coreference relations and external event knowledge.
- IDRR-Con (Shi and Demberg 2019a): a neural IDRR model attempting to learn better argument representations by leveraging the inserted connectives.
- KANN (Guo et al. 2020): a knowledge-enhanced attentive neural network that effectively fuses discourse context and its external knowledge from WordNet.
- MTL-MLoss (Nguyen et al. 2019): it is a neural model based on multi-task learning, where the labels and connectives are simultaneously predicted, with mapping losses to leverage the dependence between them.

³https://huggingface.co/transformers/model_doc/roberta.html

- BERT-FT (Kishimoto, Murawaki, and Kurohashi 2020): it fine-tunes BERT (Devlin et al. 2019) for IDRR with texts tailored to discourse classification, explicit connective prediction and implicit connective prediction.
- HierMTN-CRF (Wu et al. 2020): a hierarchical multi-task neural network stacked with a CRF layer to infer the sequence of multi-level labels.
- BMGF-RoBERTa (Liu et al. 2020): a RoBERTa-based model, which is equipped with a bilateral matching module to capture the interaction between arguments, and a gated fusion module to derive final representations for label classification.
- MTL-MLoss-RoBERTa and HierMTN-CRF-RoBERTa: for a fair comparison, we enhance MTL-MLoss and HierMTN-CRF with RoBERTa instead of ELMo and BERT, respectively.
- OurEncoder+OurDecoder: a variant of our model where the mutual learning enhanced training is removed.

Effect of the Coefficient λ

As shown in Equation 6, the coefficient λ is an important hyper-parameter that controls the relative impacts of two different losses. Thus, we vary λ from 0 to 2.5 with an increment of 0.5 each step, and inspect the performance of our model using different λ s on the validation set. The larger λ indicates that ours and the auxiliary decoder learn more knowledge from each other.

Figure 3 shows the experimental results. We can find that, compared with the model without mutual learning ($\lambda = 0$), the performance of our model at any level is always improved via mutual learning. When λ exceeding 1.0, the performance of our model tends to be stable and declines finally. Only one exception is F_1 of connective classification, which reaches its best score when $\lambda = 0.5$. Thus, we directly set $\lambda = 1.0$ for all experiments thereafter.

Main Results

Table 1 shows the main results, from which we can reach the following conclusions. **First**, all models enhanced with contextualized word embeddings (Parts 2 and 3) outperform those using static word embeddings (Part 1). **Second**, in

Model	Embeddings	Top-level		Second-level		Connective	
		F_1	Acc	F_1	Acc	F_1	Acc
IDRR-C&E (Dai and Huang 2019)	GloVe	50.49	58.32	32.13	46.03	-	-
IDRR-Con (Shi and Demberg 2019a)	word2vec	46.40	61.42	-	47.83	-	-
KANN (Guo et al. 2020)	GloVe	47.90	57.25	-	-	-	-
IDRR-C&E (Dai and Huang 2019)	ELMo	52.89	59.66	33.41	48.23	-	-
MTL-MLoss (Nguyen et al. 2019)	ELMo	53.00	-	-	49.95	-	-
BERT-FT (Kishimoto, Murawaki, and Kurohashi 2020)	BERT	58.48	65.26	-	54.32	-	-
HierMTN-CRF (Wu et al. 2020)	BERT	55.72	65.26	33.91	52.34	10.37	30.00
BMGF-RoBERTa (Liu et al. 2020)	RoBERTa	<u>63.39</u>	69.06	-	58.13	-	-
MTL-MLoss-RoBERTa	RoBERTa	61.89	68.42	38.10	57.72	7.75	29.57
HierMTN-CRF-RoBERTa	RoBERTa	62.02	<u>70.05</u>	<u>38.28</u>	<u>58.61</u>	<u>10.45</u>	<u>31.30</u>
OurEncoder+OurDecoder	RoBERTa	62.93	70.66	39.71	59.59	10.67	31.54
LDSGM	RoBERTa	63.73	71.18	40.49	60.33	10.68	32.20

Table 1: Experimental results on PDTB. The best results of previous baselines are underlined.

Model	Top-Sec	Top-Sec-Con
HierMTN-CRF	46.29	19.15
BMGF-RoBERTa	47.06	21.37
OurEncoder+OurDecoder	57.86	25.31
LDSGM	58.61	26.85

Table 2: Comparison with recent models on the consistency among multi-level label predictions. We run the code of BMGF-RoBERTa (Liu et al. 2020) and report the results.

most cases, jointly inferring multi-level labels (HierMTN-CRF-RoBERTa, OurEncoder+OurDecoder, LDSGM) performs better than separately predicting in BMGF-RoBERTa, which means that integrating the label dependence is indeed helpful. **Third**, our LDSGM model achieves the state-of-the-art performance on classification tasks at all three levels. When considering accuracy, it obtains 1.13%, 1.72% and 0.90% improvements over the best results of previous baselines (Part 3), respectively. In terms of F_1 , it also performs consistently better than previous models.

Consistency among Multi-level Label Predictions Unlike most previous studies predicting labels at different levels independently, our LDSGM model converts multi-level IDRR into a label sequence generation task, which intuitively is able to alleviate the inconsistency among label predictions at different levels. To verify this, we report the model performance via two new metrics: 1) **Top-Sec**: the percentage of correct predictions at both the top-level and second-level labels; 2) **Top-Sec-Con**: the percentage of correct predictions across all three levels. As shown in Table 2, compared with baselines, our model achieves more consistent predictions in terms of both Top-Sec and Top-Sec-Con.

Overall, all above-mentioned results strongly demonstrate the effectiveness of our model in capturing the dependence between hierarchically structured labels.

Ablation Study and Analysis

To evaluate the effects of different components, we compare LDSGM with its variants: 1) **w/o GCN**. In this variant, the

GCN is not used to learn better label embeddings, which means that the label dependence is not exploited by our encoder. 2) **w/o LA**. We replace the level-specific label attention with the one without label embeddings, where the hidden states of decoder are used as query to retrieve contexts for label predictions; 3) **w/o PP**. In this variant, the previous predictions are not used to guide the prediction of labels at the current level, that is to say, $g(\vec{y}_{m-1})$ is removed from Equation 5; 4) **w/o ML**. Mutual learning enhanced training is not involved, meaning that our model degenerates into the OurEncoder+OurDecoder model.

From Part 1 of Table 3, we can observe that our LDSGM model always exhibits better performance than their corresponding variants across all three levels. Therefore, we confirm that incorporating label dependence into both our encoder and decoder is indeed beneficial. Moreover, as shown in Part 2, with the same encoder, our LDSGM model always achieves better performance than those based on other decoding schemes, including Softmax+MultiTask, CRF, and Ensemble(OurDecoder, AuxDecoder). These results strongly demonstrate the effectiveness of exploiting the label dependence in two different directions (top-down and bottom-up). Finally, we should note that the ensemble scheme yields poor performance on connective classification. The reason behind is that, for the auxiliary decoder, it is very hard to predict connectives without the guidance of predicted higher-level labels, with only 6.79% F_1 score and 27.90% Acc.

Performance on Minority Label Predictions

Note that recent studies usually ignore the problem of addressing the highly skewed distribution of labels. For example, only about 6.8% of PDTB instances are annotated as five of the considered 11 second-level labels (*Temp.Synchrony*, *Cont.Pragmatic cause*, *Comp.Concession*, *Expa.Alternative*, and *Expa.List*). Therefore, we report the label-wise F_1 scores for the second-level labels in Table 4. A closer look into the results reveals that though our LDSGM model outperforms BMGF-RoBERTa (Liu et al. 2020) on most majority labels, the F_1 scores for three minority labels are still 0%. Besides, the BERT-based model (Kishimo-

Model	Top-level		Second-level		Connective	
	F_1	Acc	F_1	Acc	F_1	Acc
LDSGM	63.73	71.18	40.49	60.33	10.68	32.20
w/o GCN	63.47	70.77	39.32	59.54	9.81	30.94
w/o LA	62.84	70.19	39.15	59.09	10.64	31.32
w/o PP	63.68	70.78	39.27	59.62	10.08	30.86
w/o ML (aka. OurEncoder+OurDecoder)	62.93	70.66	39.71	59.59	10.67	31.54
OurEncoder+Softmax+MultiTask	62.82	70.25	38.28	58.37	9.60	30.82
OurEncoder+CRF	62.82	70.53	38.82	59.09	10.41	31.24
OurEncoder+Ensemble(OurDecoder, AuxDecoder)	63.04	70.80	39.91	59.61	8.61	29.95

Table 3: Ablation study. LA means the level-specific label attention mechanism, PP represents previous predictions, and ML means the mutual learning enhanced training. Softmax+MultiTask denotes that we stack softmax layers on the top of our encoder to individually predict labels at different levels, and then train them based on multi-task learning. Ensemble(OurDecoder, AuxDecoder) means that we directly ensemble the prediction results of ours and the auxiliary decoder.

Second-level Label	BMGF-RoBERTa	LDSGM
<i>Temp.Aynchronous</i>	56.18	56.47
<i>Temp.Synchrony</i>	0.0	0.0
<i>Cont.Cause</i>	59.60	64.36
<i>Cont.Pragmatic cause</i>	0.0	0.0
<i>Comp.Contrast</i>	59.75	63.52
<i>Comp.Concession</i>	0.0	0.0
<i>Expa.Conjunction</i>	60.17	57.91
<i>Expa.Instantiation</i>	67.96	72.60
<i>Expa.Restatement</i>	53.83	58.06
<i>Expa.Alternative</i>	60.00	63.46
<i>Expa.List</i>	0.0	8.98

Table 4: Label-wise F_1 scores for the second-level labels.

to, Murawaki, and Kurohashi 2020) also shows poor performance on minority labels. This is because that small numbers of training examples are insufficient to optimize the huge amounts of parameters in these models. In the future, we will pay more attention to the minority labels.

Related Work

Early studies for IDRR mainly resort to manually-designed features (Lin, Kan, and Ng 2009; Park and Cardie 2012; Rutherford and Xue 2014). With the rapid development of deep learning, neural network based IDRR models have gradually become dominant. In this aspect, many efforts have been devoted to learning better semantic representations of arguments, where typical studies include the shallow CNN (Zhang et al. 2015), entity-enhanced recursive neural network (Ji and Eisenstein 2015), CNN with character-enhance embeddings (Qin, Zhang, and Zhao 2016), variational generation model (Zhang et al. 2016), knowledge-augmented LSTM (Kishimoto, Murawaki, and Kurohashi 2018), and TransS-driven joint learning framework (He et al. 2020). Recently, more researchers focus on developing neural networks to better capture the semantic interaction between two input arguments for IDRR, for example, multi-level attention mechanisms (Liu and Li 2016), bi-attention mechanisms (Lei et al. 2017), topic tensor networks (Xu

et al. 2019), the knowledge-based memory network (Guo et al. 2020), and the graph-based context model (Zhang et al. 2021). To enrich the limited training data, another line of research focuses on leveraging explicit discourse data via data selection methods (Wu et al. 2016; Xu et al. 2018), multi-task learning (Liu et al. 2016; Lan et al. 2017), task-specific embeddings (Braud and Denis 2016; Wu et al. 2017), or pre-training methods (Kishimoto, Murawaki, and Kurohashi 2020). More recently, contextualized representations learned from pre-trained models, such as ELMo (Peters et al. 2018), BERT, and RoBERTa, have significantly improved the performance of IDRR (Bai and Zhao 2018; Shi and Demberg 2019b; Liu et al. 2020). All the above studies predict labels at different levels independently. Unlike them, our work infers multi-level labels jointly via a sequence generation model. The most related to ours is (Wu et al. 2020), which adopts a CRF-based decoder to infer label sequences. By contrast, our work is the first attempt to recast multi-level IDRR as a conditional label sequence generation task.

Our work is also related to recent efforts on multi-label text classification, where the dependence between labels are used to improve the encoder for better feature extraction (Xie and Xing 2018; Huang et al. 2019; Zhou et al. 2020), or the decoder for joint label predictions (Yang et al. 2018; Wu, Xiong, and Wang 2019). Ours significantly differs from the above work in following aspects: 1) Both the encoder and decoder in our model can effectively leverage the label dependence; 2) A bottom-up auxiliary decoder is introduced to boost our top-down decoder via mutual learning.

Conclusion

In this paper, we have presented a label dependence-aware sequence generation model, which can fully leverage the dependence between hierarchically structured labels for multi-level IDRR. Experimental results and in-depth analyses on the PDTB corpus clearly demonstrate the effectiveness of our model. To our knowledge, our work is the first attempt to deal with this task in the way of sequence generation. An interesting future work is to combine advantages of ours and auxiliary decoder via asynchronous bidirectional decoding (Zhang et al. 2018a; Su et al. 2019) during inference.

Acknowledgments

We would like to thank all the reviewers for their constructive and helpful suggestions on this paper. This work was supported by the National Natural Science Foundation of China (Nos. 61866012, 62172160, 62166018 and 62062034), the Natural Science Foundation of Fujian Province of China (No. 2020J06001), the Youth Innovation Fund of Xiamen (No. 3502Z20206059), and the Natural Science Foundation of Jiangxi Province of China (Nos. 20212ACB212002, 20202BABL202043).

References

- Bai, H.; and Zhao, H. 2018. Deep Enhanced Representation for Implicit Discourse Relation Recognition. In *Proceedings of COLING 2018*, 571–583.
- Braud, C.; and Denis, P. 2016. Learning Connective-based Word Representations for Implicit Discourse Relation Identification. In *Proceedings of EMNLP 2016*, 203–213.
- Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations using RNN EncoderDecoder for Statistical Machine Translation. In *Proceedings of EMNLP 2014*, 1724–1734.
- Cohan, A.; Dernoncourt, F.; Kim, D. S.; Bui, T.; Kim, S.; Chang, W.; and Goharian, N. 2018. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. In *Proceedings of NAACL 2018*, 615–621.
- Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural Language Processing (Almost) from Scratch. *The Journal of Machine Learning Research*, 12(1): 2493–2537.
- Dai, Z.; and Huang, R. 2019. A Regularization Approach for Incorporating Event Knowledge and Coreference Relations into Neural Discourse Parsing. In *Proceedings of EMNLP-IJCNLP 2019*, 2967–2978.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL 2019*, 4171–4186.
- Guo, F.; He, R.; Dang, J.; and Wang, J. 2020. Working Memory-Driven Neural Networks with a Novel Knowledge Enhancement Paradigm for Implicit Discourse Relation Recognition. In *Proceeding of AAAI 2020*, 10–18.
- He, R.; Wang, J.; Guo, F.; and Han, Y. 2020. TransS-Driven Joint Learning Architecture for Implicit Discourse Relation Recognition. In *Proceedings of ACL 2020*, 139–148.
- Huang, W.; Chen, E.; Liu, Q.; Chen, Y.; Huang, Z.; Liu, Y.; Zhao, Z.; Zhang, D.; and Wang, S. 2019. Hierarchical Multi-label Text Classification: An Attention-based Recurrent Network Approach. In *Proceedings of CIKM 2019*, 1051–1060.
- Ji, Y.; and Eisenstein, J. 2015. One Vector is Not Enough: Entity-Augmented Distributed Semantics for Discourse Relations. *Transactions of the Association for Computational Linguistics*, 3: 329–344.
- Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of ICLR 2015*.
- Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceeding of ICLR 2017*.
- Kishimoto, Y.; Murawaki, Y.; and Kurohashi, S. 2018. A Knowledge-Augmented Neural Network Model for Implicit Discourse Relation Classification. In *Proceedings of COLING 2018*, 584–595.
- Kishimoto, Y.; Murawaki, Y.; and Kurohashi, S. 2020. Adapting BERT to Implicit Discourse Relation Classification with a Focus on Discourse Connectives. In *Proceedings of LREC 2020*, 1152–1158.
- Lan, M.; Wang, J.; Wu, Y.; Niu, Z.-Y.; and Wang, H. 2017. Multi-task Attention-based Neural Networks for Implicit Discourse Relationship Representation and Identification. In *Proceedings of EMNLP 2017*, 1310–1319.
- Lei, W.; Wang, X.; Liu, M.; Ilievski, I.; He, X.; and Kan, M. Y. 2017. SWIM: A Simple Word Interaction Model for Implicit Discourse Relation Recognition. In *Proceedings of IJCAI 2017*, 4026–4032.
- Li, Y.; Feng, W.; Sun, J.; Kong, F.; and Zhou, G. 2014. Building Chinese Discourse Corpus with Connective-driven Dependency Tree Structure. In *Proceedings of EMNLP 2014*, 2105–2114.
- Lin, Z.; Kan, M.-Y.; and Ng, H. T. 2009. Recognizing Implicit Discourse Relations in the Penn Discourse Treebank. In *Proceedings of EMNLP 2009*, 343–351.
- Liu, X.; Ou, J.; Song, Y.; and Jiang, X. 2020. On the Importance of Word and Sentence Representation Learning in Implicit Discourse Relation Classification. In *Proceeding of IJCAI 2020*, 3830–3836.
- Liu, Y.; and Li, S. 2016. Recognizing Implicit Discourse Relations via Repeated Reading: Neural Networks with Multi-Level Attention. In *Proceedings of EMNLP 2016*, 1224–1233.
- Liu, Y.; Li, S.; Zhang, X.; and Sui, Z. 2016. Implicit Discourse Relation Classification via Multi-Task Neural Networks. In *Proceedings of AAAI 2016*, 2750–2756.
- Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692*.
- Nguyen, L. T.; Ngo, L. V.; Than, K.; and Nguyen, T. H. 2019. Employing the Correspondence of Relations and Connectives to Identify Implicit Discourse Relations via Label Embeddings. In *Proceedings of ACL 2019*, 4201–4207.
- Park, J.; and Cardie, C. 2012. Improving Implicit Discourse Relation Recognition Through Feature Set Optimization. In *Proceedings of SIGDIAL 2012*, 108–112.
- Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep Contextualized Word Representations. In *Proceedings of NAACL 2018*, 2227–2237.
- Prasad, R.; Dinesh, N.; Lee, A.; Miltsakaki, E.; Robaldo, L.; Joshi, A.; and Webber, B. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of LREC 2008*, 2961–2968.

- Qin, L.; Zhang, Z.; and Zhao, H. 2016. Implicit Discourse Relation Recognition with Context-aware Character-enhanced Embeddings. In *Proceedings of COLING 2016*, 1914–1924.
- Qin, L.; Zhang, Z.; Zhao, H.; Hu, Z.; and Xing, E. 2017. Adversarial Connective-exploiting Networks for Implicit Discourse Relation Classification. In *Proceedings of ACL 2017*, 1006–1017.
- Rutherford, A.; and Xue, N. 2014. Discovering Implicit Discourse Relations Through Brown Cluster Pair Representation and Coreference Patterns. In *Proceedings of EACL 2014*, 645–654.
- Shi, W.; and Demberg, V. 2019a. Learning to Explicate Connectives with Seq2Seq Network for Implicit Discourse Relation Classification. In *Proceedings of the 13th International Conference on Computational Semantics*, 188–199.
- Shi, W.; and Demberg, V. 2019b. Next Sentence Prediction helps Implicit Discourse Relation Classification within and across Domains. In *Proceedings of EMNLP 2019*, 5790–5796.
- Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(1): 1929–1958.
- Su, J.; Zhang, X.; Lin, Q.; Qin, Y.; Yao, J.; and Liu, Y. 2019. Exploiting Reverse Target-side Contexts for Neural Machine Translation via Asynchronous Bidirectional Decoding. *Artificial Intelligence*, 277: 103168.
- Tang, J.; Lin, H.; Liao, M.; Lu, Y.; Han, X.; Sun, L.; Xie, W.; and Xu, J. 2021. From Discourse to Narrative: Knowledge Projection for Event Relation Extraction. In *Proceedings of ACL 2021*, 732–742.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All You Need. In *Proceeding of NIPS 2017*, 6000–6010.
- Wu, C.; Hu, C.; Li, R.; Lin, H.; and Su, J. 2020. Hierarchical Multi-task Learning with CRF for Implicit Discourse Relation Recognition. *Knowledge-Based Systems*, 105637.
- Wu, C.; Shi, X.; Chen, Y.; Huang, Y.; and Su, J. 2016. Bilingually-constrained Synthetic Data for Implicit Discourse Relation Recognition. In *Proceedings of EMNLP 2016*, 2306–2312.
- Wu, C.; Shi, X.; Chen, Y.; Su, J.; and Wang, B. 2017. Improving Implicit Discourse Relation Recognition with Discourse-specific Word Embeddings. In *Proceedings of ACL 2017*, 269–274.
- Wu, J.; Xiong, W.; and Wang, W. Y. 2019. Learning to Learn and Predict: A Meta-Learning Approach for Multi-Label Classification. In *Proceedings of EMNLP 2019*, 4354–4364.
- Xie, P.; and Xing, E. 2018. A Neural Architecture for Automated ICD Coding. In *Proceedings of ACL 2018*, 1066–1076.
- Xu, S.; Li, P.; Kong, F.; Zhu, Q.; and Zhou, G. 2019. Topic Tensor Network for Implicit Discourse Relation Recognition in Chinese. In *Proceedings of ACL 2019*, 608–618.
- Xu, Y.; Hong, Y.; Ruan, H.; Yao, J.; Zhang, M.; and Zhou, G. 2018. Using Active Learning to Expand Training Data for Implicit Discourse Relation Recognition. In *Proceedings of EMNLP 2018*, 725–731.
- Yang, P.; Sun, X.; Li, W.; Ma, S.; Wu, W.; and Wang, H. 2018. SGM: Sequence Generation Model for Multi-label Classification. In *Proceedings of COLING 2018*, 3915–3926.
- Zhang, B.; Su, J.; Xiong, D.; Lu, Y.; Duan, H.; and Yao, J. 2015. Shallow Convolutional Neural Network for Implicit Discourse Relation Recognition. In *Proceedings of EMNLP 2015*, 2230–2235.
- Zhang, B.; Xiong, D.; Su, J.; Liu, Q.; Ji, R.; Duan, H.; and Zhang, M. 2016. Variational Neural Discourse Relation Recognizer. In *Proceedings of EMNLP 2016*, 382–391.
- Zhang, X.; Su, J.; Qin, Y.; Liu, Y.; Ji, R.; and Wang, H. 2018a. Asynchronous Bidirectional Decoding for Neural Machine Translation. In *Proceedings of AAAI 2018*, 5698–5705.
- Zhang, Y.; Meng, F.; Li, P.; Jian, P.; and Zhou, J. 2021. Context Tracking Network: Graph-based Context Modeling for Implicit Discourse Relation Recognition. In *Proceedings of NAACL 2021*, 1592–1599.
- Zhang, Y.; Xiang, T.; Hospedales, T. M.; and Lu, H. 2018b. Deep Mutual Learning. In *Proceeding of CVPR 2018*, 4320–4328.
- Zhou, J.; Ma, C.; Long, D.; Xu, G.; Ding, N.; Zhang, H.; Xie, P.; and Liu, G. 2020. Hierarchy-Aware Global Model for Hierarchical Text Classification. In *Proceedings of ACL 2020*, 1106–1117.