DiscoPrompt: Path Prediction Prompt Tuning for Implicit Discourse Relation Recognition

Chunkit Chan[*1], **Xin Liu**[*1], **Jiayang Cheng**[1], **Zihan Li**[1], **Yangqiu Song**[1],
Ginny Y. Wong[2], **Simon See**[2]
[1]Department of Computer Science and Engineering, HKUST, Hong Kong SAR, China
[2]NVIDIA AI Technology Center (NVAITC), NVIDIA, Santa Clara, USA
{ckchancc, xliucr, jchengaj, zliho, yqsong}@cse.ust.hk
{gwong, ssee}@nvidia.com

Abstract

Implicit Discourse Relation Recognition (IDRR) is a sophisticated and challenging task to recognize the discourse relations between the arguments with the absence of discourse connectives. The sense labels for each discourse relation follow a hierarchical classification scheme in the annotation process (Prasad et al., 2008), forming a hierarchy structure. Most existing works do not well incorporate the hierarchy structure but focus on the syntax features and the prior knowledge of connectives in the manner of pure text classification. We argue that it is more effective to predict the paths inside the hierarchical tree (e.g., "*Comparison -> Contrast -> however*") rather than flat labels (e.g., *Contrast*) or connectives (e.g., *however*). We propose a prompt-based path prediction method to utilize the interactive information and intrinsic senses among the hierarchy in IDRR. This is the first work that injects such structure information into pre-trained language models via prompt tuning, and the performance of our solution shows significant and consistent improvement against competitive baselines.

1 Introduction

Discourse parsing is the task of automatically parsing discourse structure in a text, including the identification of discourse structure and the annotation of discourse relations (Li et al., 2022). Discourse Relation Recognition (DRR) is a crucial task in discourse parsing, recognizing relations between two arguments (i.e., sentences or clauses). It is vital for textual coherence and is considered as the essential step for many downstream tasks involving more context, such as question answering (Rutherford and Xue, 2015), text generation (Bosselut et al., 2018), and argument mining (Liu et al., 2021b). Explicit discourse relation recognition (EDRR) has already been demonstrated that utilizing explicit connectives information can effectively determine the discourse relation types (Varia et al., 2019). On the other hand, implicit discourse relation recognition (IDRR) is still challenging with the absence of connectives (Varia et al., 2019).

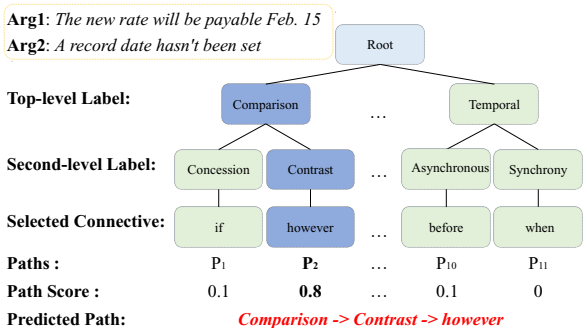

Figure 1: An example of the implicit discourse relation hierarchy and path prediction.

Traditional works on IDRR focus on syntax features, including word pairs (Lin et al., 2009; Varia et al., 2019) and other surface features (Ji and Eisenstein, 2015; Bai and Zhao, 2018). With deep neural networks and large language models (LLMs), different approaches pay much attention to text representations via attention (Liu and Li, 2016), pre-training (Shi and Demberg, 2019b), multi-task learning (He et al., 2020; Long and Webber, 2022), and prior knowledge (Liu et al., 2020; Zhou et al., 2022). But one important piece of information, i.e., the inherent discourse label hierarchy, is not fully investigated.

The sense labels for each discourse relation follow a hierarchical classification scheme in the annotation process of PDTB 2.0 framework (Prasad et al., 2008), forming a hierarchy structure. Figure 1 shows an example from PDTB 2.0 dataset (Prasad et al., 2008). It consists of two arguments (i.e., Arg1 and Arg2) and is annotated with relation senses, where the semantics of the top-level *Comparison* is further refined by the second-level *Contrast*. Besides, we list representative connectives (e.g., *however*) to help better understand the definitions and semantics of labels. LDSGM (Wu et al., 2022) uses

[*] Equal contribution.

Findings of the Association for Computational Linguistics: ACL 2023, pages 35–57
July 9-14, 2023

graph convolutional networks to encode the label dependencies into text representations, illustrating the importance of label structures on text representation learning and label prediction. However, such usage is not compatible with pre-training because it may significantly affect the representations from language models. Prompt tuning has shown its power in text classification without altering the representations from pre-trained language models, especially for low-resource scenarios (Schick and Schütze, 2021; Gao et al., 2021).

In this paper, we propose a prompt-based path prediction method, **Disco**urse relation path prediction **Prompt** tuning model (**DiscoPrompt** [1]), to utilize the hierarchy and intrinsic senses of labels in IDRR. Specifically, we transform the hierarchy in Figure 1 to "Comparison -> Concession -> if; $\cdots$; Temporal -> Synchrony -> when" as the hierarchical prompt and add it as the prefix of arguments to be classified. The dependencies of top and second-level relation senses are explicitly provided as the context. On the other hand, connectives are provided as the natural language explanations of labels to help the language models better adapt to the prior knowledge. We ask the LLMs to predict the label's hierarchical path instead of the leaf label for IDDR, and we show such a way of providing the label hierarchy ahead of arguments significantly improves the IDRR performance. Our contributions are summarized as follows:

- This is the first work that injects labels' hierarchical structure information and connectives into pre-trained language models via prompt tuning.
- We model the IDRR problem as the path prediction problem that predicts the joint probability of top-level relations, second-level types, and connectives at the same time.
- We conduct extensive experiments and thorough ablation studies to discuss the necessity and effectiveness of the label hierarchy and connectives. The results support our claims and the success of our proposed DiscoPrompt model.

2 Related Work

Prompt Tuning With LLMs, such as T5 (Raffel et al., 2020) and GPT-3 (Brown et al., 2020), prompt-based methods have attracted much attention in the field of natural language understanding (Schick and Schütze, 2021; Lester et al., 2021; Liu et al., 2022). Compared with fine-tuning, prompt tuning may have a better generalization on various tasks due to the aligned nature of language descriptions and answer semantics, e.g., classification problems (Gao et al., 2021; Wang et al., 2022a). At the same time, there are some efforts to leverage prompts with structural inputs for knowledge customization (Zhong et al., 2022). Injecting hierarchy information into prompts is also promising. For example, using top-level predictions to refine prompts of bottom levels can surpass soft prompts and hard prompts (Wang et al., 2022b). Nevertheless, how to employ LLMs to better involve hierarchy knowledge is still under investigation.

Implicit Discourse Relation Recognition It has been discovered that connectives can provide necessary clues in predicting discourse relations to achieve around 95% accuracy (Dai and Huang, 2019; Varia et al., 2019). However, the absence of connectives makes the prediction more challenging. Many efforts have been paid to explore the syntax through linguistic features (Rutherford and Xue, 2015; Ji and Eisenstein, 2015; Wang and Lan, 2016; Dai and Huang, 2018; Varia et al., 2019), attention (Liu and Li, 2016; Bai and Zhao, 2018), pre-training (Shi and Demberg, 2019b), knowledge transfer (Lan et al., 2017; Dai and Huang, 2019; He et al., 2020), etc. With the power of language models, connective prediction also illustrates its effectiveness in implicit relation prediction (Nguyen et al., 2019; Shi and Demberg, 2019a; Kishimoto et al., 2020; Kurfali and Östling, 2021). In addition, PCP (Zhou et al., 2022) shows the feasibility of combining label prediction and connective prediction under the manner of prompts. The latest methods reveal the significance of the label hierarchy of discourse relations. LDSGM (Wu et al., 2022) utilizes the graph convolutional networks to incorporate label dependencies into text representations, while ContrastiveIDRR (Long and Webber, 2022) leverages the sense hierarchy to obtain contrastive learning representation. However, these methods are incompatible with pre-training as they modify the representations from pre-trained language models. Therefore, this work investigates injecting the label dependencies information and connectives into pre-trained language models via prompt tuning with aligning the representations.

[1]The source code is available at https://github.com/HKUST-KnowComp/DiscoPrompt

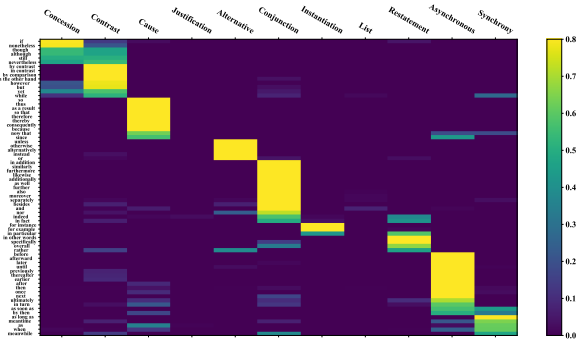

Figure 2: Prior probabilities of PDTB 2.0 frequent connectives.

Top-level	Second-level	Connectives
Comparison	Concession Contrast	if however
Contingency	Cause Justification	so indeed
Expansion	Alternative Conjunction Instantiation List Restatement	instead also for example and specifically
Temporal	Asynchronous Synchrony	before when

Table 1: The label word set on PDTB 2.0 dataset, includes four top-level relations, 11 second-level subtypes, and 11 connectives.

3 Method

3.1 Problem Definition

The sense labels in various levels of the Implicit Discourse Relation Recognition (IDRR) task naturally constitute a hierarchy, denoted as $\mathcal{H}$. $\mathcal{H}$ is a hierarchical tree structure whose depth is d, with the root node in depth 0 and class sense of different levels distributed to the corresponding layer (i.e., from depth 1 to d) in this tree. Let class label set $\mathcal{C}$ to be $\bigcup_{k=1}^{d} \mathcal{C}^k$ where $\mathcal{C}^k = \{c_1^k, \cdots, c_{n_k}^k\}$ is the label set of depth k, and n_k is the number of classes at depth k. For example, the hierarchy $\mathcal{H}$ of PDTB 2.0 forms a tree with depth size 2, and the $\mathcal{C}^2$ corresponds to the label set of the second level, containing 11 class subtypes like *Concession*, *Synchrony*, etc. We can enrich the label hierarchy by adding a connective layer like in Figure 1. We adopt the Naive Bayes to compute the prior distribution $\Pr(c^2|z)$ from the explicit relation data, where $c^2 \in \mathcal{C}^2$ is a subtype, and z is the connectives. Figure 2 shows the heat map of highly frequent connectives. We can find that the connectives are the vital clue for discourse relations. Therefore, we select the most discriminative ones as $\mathcal{C}^3$. We do not observe significant improvement when adding more than one connective for each c^2. Therefore, we summarize $\mathcal{C}$ for PDTB 2.0 in Table 1. Prior distributions and label words of CoNLL16 are shown in Appendix A.2.

In this task, given a data set $\mathcal{D} = \{(x_i, y_i)\}$ consisting of data instance $x_i = (a_i^1, a_i^2)$ and label y_i, where the a_i^1, a_i^2 represent the argument 1 and argument 2 of respective instance i and the label y_i is class label set. In our method, the class label set including d labels for d layers forms a path $\mathcal{P}$ in hierarchical tree $\mathcal{H}$, instead of a single class label for a specific level. After predicting a path, the classes of various levels are the nodes lying in the predicted path. Therefore, this task is to find out the optimal path:

$$P_i^* = \arg\max_{\mathcal{P}^j} \Pr\left(\mathcal{P}^j \mid x_i\right), \quad (1)$$

where P_i^* is the optimal path and j indicates the j-th path among all paths.

3.2 T5 Backbone Model

T5 (Raffel et al., 2020) is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks. The unsupervised denoising training task required the model only to predict the masked consecutive spans of tokens. For example, the input "Thank you for inviting me to your party last week." will be corrupted as "Thank you <X> me to your party <Y> week." and the target is " <X> for inviting <Y> last </s>" </s> is the eos_token. In the supervised pre-trained task, the model was asked to perform the sequence-to-sequence input-output mapping by specifying the task prefix (such as "translate German to English:" or "summarize:"). However, the specific textual prefix token is difficult to discover and requires a substantial amount of human effort. Hence, prefix tuning (Li and Liang, 2021) and prompt tuning (Lester et al., 2021) methods proposed to overcome this problem by relaxing the constraint of discrete textual tokens to continuous tunable ones.

3.3 Path Prediction Prompt Tuning Method

To predict the path P_i^* for each instance x_i, we leverage a human-tailored template $\mathcal{T}(\cdot)$ to convert the data instances to the prompt input $\tilde{x}_i = \mathcal{T}(x_i)$ and a verbalizer $\mathcal{V}(\cdot)$ to map a set of words to class labels. Figure 3 illustrates the architecture of **DiscoPrompt**.

Structure-Aware Prompt The crafted template includes necessary discrete tokens, masked tokens, soft continuous tokens, and context with the hierarchy information. The first part of our prompt

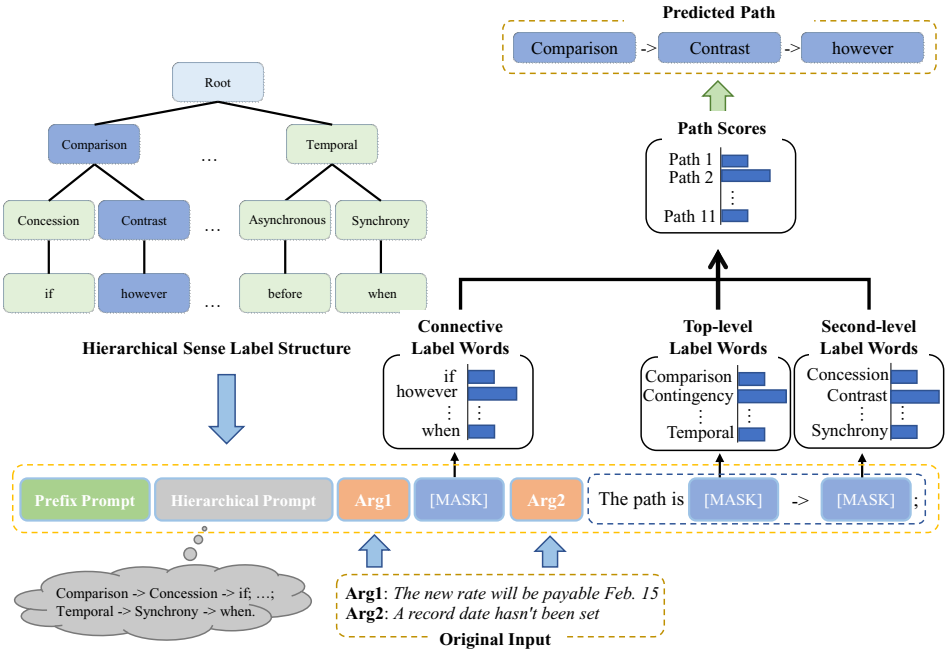

Figure 3: DiscoPrompt model architecture.

template is the discrete tokens "The path is " for eliciting the predicted path $\mathcal{P}_i$. Then three [MASK] tokens are included: an [MASK] is inserted between two arguments for predicting the probability of decided connectives, and two [MASK]s form an edge "[MASK] -> [MASK]" for receiving the top and second-level class probabilities. We also added 20 learnable continuous tokens at the beginning of the template to effectively searching an optimal template. To better utilize the hierarchy information and senses of labels, we explicitly translate them into a tailored hierarchical tree prompt and insert it into the input. This hierarchical tree prompt is the discrete tokens appended ahead of the arguments as the context in natural language. Figure 8 in Appendix B.2 shows the details of the template.

Path Verbalizer A traditional verbalizer usually maps a label y to a single answer token z or a series of spans $z^1, z^2, \cdots$ greedily (Schick and Schütze, 2021; Liu et al., 2021a). We extend it by mapping a path $\mathcal{P}$ to three tokens, i.e. $\{\mathcal{P}^j\} \rightarrow \mathcal{Z} \times \mathcal{Z} \times \mathcal{Z}$, where $\mathcal{Z}$ is the vocabulary. We denote the three [MASK] tokens as z^1, z^2, and z^3. Then using the prompt template with three [MASK]s and the verbalizer $\mathcal{V}(\cdot)$, the probability distribution over $\{\mathcal{P}^j\}$ can be formalized as the joint probabilities of z^1, z^2, and z^3, i.e. $\Pr(\mathcal{P}^j \mid \tilde{x}_i) = \Pr(\mathcal{V}(\mathcal{P}^j) \mid \tilde{x}_i) = \Pr(z_i^1 = p_3^j, z_i^2 = p_1^j, z_i^3 = p_2^j \mid \tilde{x}_i)$, where a path $\mathcal{P}^j$ consists of p_1^j (the top-level), p_2^j (the second-level), and p_3^j (the connective). Since T5 can synchronously predict masked tokens, the joint probability can be written as

$$\Pr(\mathcal{P}^j \mid \tilde{x}_i) = \prod_{k=1}^{3} \Pr(z_i^k = v^k(\mathcal{P}^j) \mid \tilde{x}_i), \quad (2)$$

where $v^k(\cdot) : \{\mathcal{P}^j\} \rightarrow \mathcal{Z}$ is the submap of $\mathcal{V}(\cdot)$ for the k-th [MASK]. The final learning objective of DiscoPrompt is to maximize

$$\mathcal{J} = \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} \log \sum_{k=1}^{3} \Pr(z_i^k = v^k(\mathcal{P}^j) \mid \tilde{x}_i). \quad (3)$$

Once we get the prediction of $\mathcal{P}_i^*$ by choosing the maximum joint probability (i.e., path score) as Eq. (2), we can get the prediction of each level as Eq. (4).

$$c_i^{k^*} = \arg\max_{c^k} \Pr(c^k \mid \mathcal{P}^j, x_i) \cdot \Pr(\mathcal{P}^j \mid x_i), \quad (4)$$

where $\Pr(c^k \mid \mathcal{P}^j, x_i)$ can be calculated by the prior probability (or simply set as 1.0).

4 Experimental Setting

4.1 Dataset

The experiments are conducted on two datasets, the PDTB 2.0 (Prasad et al., 2008) and the CoNLL-2016 shared task (CoNLL16) (Xue et al., 2016), to validate the performance of our method. Both contain the Wall Street Journal (WSJ) articles, and the difference is the annotation and relation senses. We evaluate performance on PDTB 2.0 according to two different settings denoted as *Ji* (Ji and Eisenstein, 2015) and *Lin* (Lin et al., 2009) with 11 subtypes. The CoNLL-2016 shared task provides more

Models	Ji (Top)		Ji (Second)		Lin (Top)		Lin (Second)	
	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy
MTL-MLoss (Nguyen et al., 2019)	53.00	-	-	49.95	-	-	-	46.48
ELMo-C&E (Dai and Huang, 2019)	52.89	59.66	33.41	48.23	-	-	-	-
RWP-CNN (Varia et al., 2019)	50.20	59.13	-	-	-	-	-	-
TransS (He et al., 2020)	-	-	-	-	51.24	59.94	-	-
BMGF-RoBERTa (Liu et al., 2020)	63.39	69.06	35.25	58.13	*58.54*	*68.66*	*39.15*	*53.96*
CG-T5 (Jiang et al., 2021)	57.18	65.54	37.76	53.13	-	-	-	-
LDSGM (Wu et al., 2022)	63.73	71.18	40.49	60.33	-	-	-	-
GOLF (Jiang et al., 2022b)	65.76	72.52	41.74	61.16	-	-	-	-
ContrastiveIDRR (Long and Webber, 2022)	67.85	71.70	45.54	59.19	-	-	-	-
XLNet (base, cased) (Kim et al., 2020)	*59.33*	*66.35*	*36.36*	*54.73*	*56.16*	*68.05*	*36.23*	*55.82*
XLNet (large, cased) (Kim et al., 2020)	*63.58*	*69.52*	*38.24*	*61.29*	*58.97*	72.17	*40.71*	*58.77*
OTMT (XLNet-base) (Jiang et al., 2022a)	60.78	68.89	-	56.65	-	-	-	56.37
OTMT (XLNet-large) (Jiang et al., 2022a)	64.46	72.34	-	61.06	-	-	-	61.62
Fine-Tuning (T5-base) (Raffel et al., 2020)	*57.61*	*65.39*	*33.96*	*55.53*	*50.50*	*63.59*	*36.49*	*51.96*
Fine-Tuning (T5-large) (Raffel et al., 2020)	*61.37*	*69.69*	*38.04*	*57.65*	*58.12*	*71.13*	*42.04*	*59.40*
Prefix-Tuning (T5-base) (Li and Liang, 2021)	*25.87*	*52.45*	*7.49*	*31.09*	*25.08*	*54.18*	*8.45*	*26.37*
Prefix-Tuning (T5-large) (Li and Liang, 2021)	*63.74*	*71.51*	*39.73*	*59.77*	*58.06*	*69.84*	*36.86*	*56.53*
Prompt-Tuning (T5-base) (Lester et al., 2021)	*30.17*	*56.11*	*15.01*	*38.21*	*25.26*	*55.09*	*8.97*	*27.68*
Prompt-Tuning (T5-large) (Lester et al., 2021)	*66.95*	*71.99*	*44.08*	*60.15*	59.92	*71.02*	*40.75*	*60.44*
PCP (RoBERTa-base) (Zhou et al., 2022)	64.95	70.84	41.55	60.54	*53.00*	*66.58*	*41.19*	*56.14*
PCP (RoBERTa-large) (Zhou et al., 2022)	67.79	73.80	44.04	61.41	*52.75*	*71.13*	43.04	*60.44*
DiscoPrompt (T5-base)	65.79	71.70	43.68	61.02	64.90	71.28	41.82	59.27
DiscoPrompt (T5-large)	**70.84**	**75.65**	**49.03**	**64.58**	**67.06**	**73.76**	**45.25**	**63.05**
DiscoPrompt (T5-11b)	75.34	78.06	52.42	68.14	72.78	77.55	47.18	67.62

Table 2: The accuracy (%) and F1 score (%) are evaluated on the PDTB 2.0 dataset. *Italics numbers* indicate the results of reproduced models, underlined numbers correspond to the second best. ContrastiveIDRR corresponds to the model without a data augmentation for a fair comparison. More baselines before 2019 can be found in Table 14 in Appendix.

abundant annotations and two test data denoted as *Test* and *Blind* with 15 subtypes. More specific details and statistics are listed in Appendix A.1.

4.2 Implementation Details

We employ the T5 model (Raffel et al., 2020) as the backbone to implement **DiscoPrompt** and use the T5-large as the primary model for a fair comparison with extensive baselines. Generally, the overall configuration follows the setting in Lester et al. (2021), and we put more details of the configuration in Appendix A.2. We report the Macro-F1 score and accuracy in experiments and ablation studies. A prediction is considered as correct whenever it matches one of the ground-truth labels. All experiments are conducted with 2 × NVIDIA V100 (32GB) except for the T5-11b scale on 2 × NVIDIA A6000 (48GB).

4.3 Baselines

This paper mainly adopts two categories of competitive baselines for the PDTB 2.0 dataset and the CoNLL-2016 shared task[2]. The first category is the previous state-of-the-art (SOTA) baselines, such as TransS (He et al., 2020), BMGF-RoBERTa (Liu et al., 2020), LDSGM (Wu et al., 2022), XLNet-large (Kim et al., 2020), OTMT (XLNet-large) (Jiang et al., 2022a), and ContrastiveIDRR (Long and Webber, 2022). Two partitions of these SOTA baselines are highlighted for comparison with our method. One partition utilizes the hierarchical information in their methods (e.g., the LDSGM and ContrastiveIDRR), and the other is to fine-tune the pre-trained language models (e.g., XLNet-large). Therefore, we include the fine-tuned T5 models to illustrate the performance gain of prompt tuning. Besides, a prompt-based method PCP (Zhou et al., 2022) and general Prefix-Tuning (Li and Liang, 2021), as well as Prompt Tuning (Lester et al., 2021) are included. The details of implementation are listed in A.3.

5 Experimental Result

5.1 Main Results

Table 2 and Table 3 summarize the main results of the PDTB 2.0 and CoNLL16 datasets, from which we derive the following conclusions. **First**, our method significantly outperforms all baselines and achieves state-of-the-art performance at both top and second-level classes in the IDRR task. Specifically, our method gains a considerable improvement of 6.93% second-level accuracy, 10.99% second-level F1 score, 5.96% top-level accuracy, and 9.47% top-level F1 score over the fine-tuning of the T5-large model in PDTB (*Ji*). It demonstrates that our method effectively utilizes the struc-

[2]We report our produced results via the official code if the authors did not report results on those data.

Models	Test (Top)		Test (Second)		Blind (Top)		Blind (Second)	
	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy
CoNLL Baseline (Rutherford and Xue, 2016)	-	-	-	36.13	-	-	-	37.67
MTL-Attn-LSTM (Lan et al., 2017)	-	-	-	39.40	-	-	-	40.12
RWP-CNN (Varia et al., 2019)	-	-	-	39.39	-	-	-	39.36
BMGF-RoBERTa (Liu et al., 2020)	*56.55*	*68.23*	40.68	57.26	*58.30*	*74.43*	28.98	55.19
XLNet (base, cased) (Kim et al., 2020)	*43.48*	*62.29*	*18.80*	*33.16*	*19.90*	*66.12*	*9.07*	*28.71*
XLNet (large, cased) (Kim et al., 2020)	*47.07*	*64.76*	*27.13*	*47.85*	*22.37*	*66.59*	*11.94*	*35.06*
Fine-Tuning (T5-base) (Raffel et al., 2020)	*54.64*	*67.10*	*31.99*	*53.92*	*50.94*	*71.30*	*24.52*	*49.89*
Fine-Tuning (T5-large) (Raffel et al., 2020)	*58.74*	*70.87*	*34.66*	*58.88*	*56.28*	*73.07*	*24.63*	*54.30*
Prefix-Tuning (T5-base) (Li and Liang, 2021)	*26.18*	*55.35*	*8.26*	*26.63*	*27.17*	*65.88*	*9.70*	*32.71*
Prefix-Tuning (T5-large) (Li and Liang, 2021)	*57.84*	*71.15*	*46.06*	*59.40*	*55.61*	*74.12*	*30.53*	*55.53*
Prompt-Tuning (T5-base) (Lester et al., 2021)	*25.53*	*54.44*	*13.01*	*29.11*	*27.21*	*64.71*	*11.55*	*33.65*
Prompt-Tuning (T5-large) (Lester et al., 2021)	*59.95*	*72.32*	*49.59*	*60.57*	*63.35*	*77.41*	*35.72*	*57.88*
PCP (RoBERTa-base) (Zhou et al., 2022)	*58.54*	*69.31*	*33.27*	*55.48*	*55.30*	*72.00*	*26.00*	*50.99*
PCP (RoBERTa-large) (Zhou et al., 2022)	*63.78*	*72.69*	*37.79*	*58.36*	*64.74*	*76.47*	*27.77*	*56.24*
DiscoPrompt (T5-base)	60.66	70.63	45.99	60.84	62.98	76.94	39.27	57.88
DiscoPrompt (T5-large)	**69.56**	**75.33**	**56.29**	**66.32**	**67.89**	**80.47**	**38.49**	**63.06**
DiscoPrompt (T5-11b)	70.38	78.07	57.75	69.71	72.33	84.94	38.60	66.35

Table 3: The accuracy (%) and F1 score (%) are evaluated on the implicit discourse partition of CoNLL16 dataset. *Italics number* indicate the results of reproduced models.

ture information and perceives the specific knowledge on the correlation of discourse relations and connectives and finally enhances the ability of T5 to undertake this challenging task. **Second**, the prompt-based baselines (e.g., Prefix-Tuning, Prompt-Tuning, and PCP) receive outstanding performance and perform better than the T5-large fine-tuning method on this task. Many works (Scao and Rush, 2021; Lester et al., 2021) have discussed the overfitting problem of T5-large fine-tuning, and this can be partially solved by prompt-tuning by updating a few learnable parameters with limited training instances. The learnable parameters of baselines and DiscoPrompt are shown in Appendix A.6. **Third**, the ContrastiveIDRR and our method obtain better F1 scores. This observation can support the necessity of integrating the dependencies among relations as well as connectives in the label hierarchy.

Fine-tuning a relatively pre-trained large language model (LLM) such as T5-11b requires extensive computation resources to update all trainable parameters. However, by adapting the prompt tuning-based method, the entire LLM is frozen, and only a few learnable parameters of input embeddings are required to update to obtain satisfactory performance. Therefore, we also include the performance of DiscoPrompt with the T5-11b version as a reference to explore the ability of a sizeable pre-trained language model on this IDDR task. As shown in Table 2 and Table 3, DiscoPrompt (T5-11b) easily beats other methods, achieving a 52.42% F1 score and 68.14% accuracy in the 11-class classification (second-level) task of the PDTB (*Ji*) and illustrating the benefits without adjusting the representations from LLMs. On the contrary, fine-tuning T5-11b is infeasible in most single compute nodes. Considering the computation cost, we still focus on the comparison among large models.

5.2 Ablation Study

To better investigate the factors of DiscoPrompt, we design numerous ablations on the path prediction and the tailored hierarchical tree prompt. Table 4 reports the performance of the ablation study for our model in the PDTB (*Ji*).

Joint Probability for Path Prediction In our method, by estimating the likelihoods of p_1^j (the top-level), p_2^j (the second-level), and p_3^j (the connective) in a predicted path, the dependencies of these three masks are utilized for enhancing the ability of the pre-trained language model on this IDRR task. According to the experimental results in Table 4, we can conclude that 1) the performance of the path prediction model incorporating the signals from all three masks surpasses other models (i.e., paths forming by two arbitrary masks or one connective mask), emphasizing the significance of dependencies and effectiveness of joint prediction; 2) the predicted path model without prior knowledge of selected discriminative connectives (i.e., Path w/ Top & Second) performs the worst, which is consistent with findings in Zhou et al. (2022); 3) the predicted path model with only the connective mask (e.g., Path w/ Connective) performs consistently worse than paths adding the second mask, indicating the slight ambiguity of connectives and the necessity of the label hierarchy especially with the top. The performance gain with the complete path is at least 3.76% on average, and models associating with paths including individual connective masks can also beat the previous SOTA.

Model		F1 (Top)	Accuracy (Top)	F1 (Second)	Accuracy (Second)
PCP (RoBERTa-large) (Zhou et al., 2022)		67.79	73.80	44.04	61.41
DiscoPrompt (T5-large)		**70.84**	**75.65**	**49.03**	**64.58**
Path	w/ Top & Second	53.93	66.89	33.74	53.71
	w/ Top & Connective	69.19	72.57	42.95	64.08
	w/ Second & Connective	70.04	74.69	45.98	64.37
	w/ Connective	68.00	73.82	43.76	63.43
	w/ Second	63.45	71.99	40.52	59.67
Prompt	w/o Entire Discrete Prompt	68.38	72.95	41.79	62.66
	w/o Cloze Discrete Prompt	68.64	73.72	41.44	63.72
	w/o Hierarchical Tree Prompt	68.03	72.18	43.14	62.85
Hierarchy	w/ Continuous Hierarchy Prompt	67.63	73.24	44.03	63.81
	w/ Continuous Labels & Connective	67.74	73.24	44.06	64.10
	w/ Continuous Connective	68.35	73.15	44.48	64.20

Table 4: Ablation study in the components of DiscoPrompt on PDTB (*Ji*). The path part considers different combinations in the path prediction; the prompt part tries to eliminate templates from the structure-aware prompt; the hierarchy replaces the hierarchical tree prompt with continuous variants.

Model	Comp.	Cont.	Exp.	Temp.
MTL-MLoss (Nguyen et al., 2019)	48.44	56.84	73.66	38.60
KANN (Guo et al., 2020)	43.92	57.67	73.45	36.33
BMGF-RoBERTa (Liu et al., 2020)	59.44	60.98	77.66	50.26
CG-T5 (Jiang et al., 2021)	55.40	57.04	74.76	41.54
CVAE (Dou et al., 2021)	55.72	63.39	80.34	44.01
ContrastiveIDRR (Long and Webber, 2022)	65.84	63.55	79.17	69.86
DiscoPrompt (T5-base)	62.55	64.45	78.77	57.41
DiscoPrompt (T5-large)	67.13	69.76	81.61	64.86
DiscoPrompt (T5-11b)	**74.35**	**72.44**	**82.57**	**72.00**

Table 5: The performance for top-level classes on PDTB (*Ji*) in terms of F1 (%) (top-level multi-class classification). More baselines for comparison can be found in Table 17 in Appendix B.3.

Second-level Label	PCP	Contrast	DP (large)	DP (11b)
Temp.Asynchronous	57.81	59.79	64.15	**72.27**
Temp.Synchrony	0.0	**78.26**	50.00	33.33
Cont.Cause	65.64	65.58	69.66	**72.28**
Cont.PragmaticCause	0.0	0.0	0.0	0.0
Comp.Contrast	63.88	62.63	62.88	**70.63**
Comp.Concession	8.00	0.0	**9.09**	0.0
Exp.Conjunction	57.78	58.35	60.09	**62.84**
Exp.Instantiation	74.01	73.04	74.17	**76.60**
Exp.Restatement	61.00	60.00	65.24	**65.98**
Exp.Alternative	66.67	53.85	60.00	**84.21**
Exp.List	29.63	34.78	24.00	**38.46**

Table 6: The label-wise F1 scores for the second-level labels on PDTB (*Ji*) (second-level multi-class classification). "Contrast" and "DP" indicate the ContrastiveIDRR and DiscoPrompt. Results of more baselines are listed in Table 19 in Appendix B.3.

Discrete Prompt Template Two portions in our designed prompt template are in natural textual form and as discrete non-tunable tokens. The first part is the discrete tokens for the label hierarchy structure (i.e., **hierarchical tree prompt**), shown in Figure 3 and Figure 8. The second part is the **cloze discrete prompt** "The path is". We remove the discrete tokens from the template to evaluate their importance. The performance shown in Table 4 demonstrates that the two parts of the prompt are essential for achieving satisfactory performance compared with the without manual tips (i.e., Prompt w/o Entire Prompt). When adding back the cloze discrete prompt, we do not observe the model's ability to understand the correlations among masks for path prediction. Without explicitly injecting structural information into the hierarchical tree prompt, the performance dropped significantly, especially the second-level F1 score, dropping from 49.03% to 43.14%.

Hierarchical Tree Prompt To acquire a deeper understanding of the discrete hierarchical tree prompt, we perform experiments to gradually replace the discrete tokens with continuous ones in various elements of this hierarchy prompt. The experiments include 1) Continuous Hierarchy Prompt: replacing the whole hierarchical tree prompt as the continuous tunable prompt with the same number of tokens, 2) Continuous Labels & Connective: only including the "-> " and replacing other relation labels and connective as continuous tunable prompt, and 3) Continuous Connective: only replacing the textual connective to be the tunable prompt. The experimental result in Table 4 underscores the importance and effectiveness of our tailored discrete hierarchical tree prompt, which obtains at least 4.98% performance boost.

Prompt Engineering Furthermore, we conduct the prompt template searching and the parameter sensitivity on the continuous prompt length that we describe in Appendix B.2.

5.3 Label-wise F1 Scores

The PDTB (*Ji*) setting exhibits highly skewed label distributions, with only roughly 854 training instances (i.e., 6.8% of 12406 training instances) annotating as five of the 11 second-level labels. To further explore our model in four top-level relations and 11 second-level sense types on this dataset, Table 5 and Table 6 report the F1 scores (%) of the top-

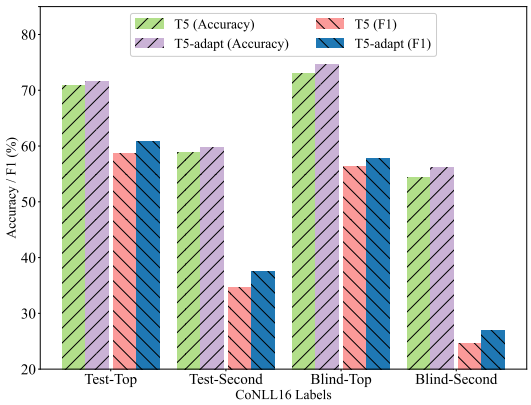

Figure 4: The performance comparison of the T5-large fine-tuning with and without using our designed template on the CoNLL16 dataset.

level and second-level classes, respectively. In Table 5, our model outperforms all baselines in three top-level relations (i.e., *Comparison*, *Contingency*, *Expansion*), and most of the baselines in the *Temporal* relation except ContrastiveIDRR. Specifically, Table 6 illustrates that our model performs better on the *Temp.Asynchronous* second-level class, whereas ContrastiveIDRR is much better on the *Temp.Synchrony*. In Table 6, our model obtains valid predictions on most second-level classes, but all methods fail to predict *Cont.Pragmatic Cause*. This situation may result from the few training examples of this class being insufficient for optimal learnable parameters, and the models tend to ignore this class in the prediction process. When we check the less representative classes (i.e., *Temp.Synchrony*, *Comp.Concession*), DiscoPrompt can still make correct predictions, while PCP and ContrastiveIDRR still fail to predict neither correct ones. Moreover, we can also see the power of LLMs that the T5-11b performs remarkably better than smaller models.

5.4 Prompt Adaptation

For T5 Fine-Tuning To demonstrate the effectiveness of our designed template and explore whether our designed template can be used for the fine-tuning paradigm, we convert the data input to the tailored prompt template but with only a [MASK] for generating the entire path. The experimental results on CoNLL16 are summarised in Figure 4, and the T5-adapt boosts all metrics over vanilla T5-large fine-tuning. The detailed performance and the experimental results for PDTB 2.0 are shown in Table 18 and Figure 9 in Appendix B.4.

5.5 Prompt Adaptation For ChatGPT

With the powerful ability of LLMs exhibited on numerous tasks, we are curious about the capability of ChatGPT on zero-shot IDRR task. We test the ability of ChatGPT with three designed templates on the PDTB (*Ji*), and the performance is shown in Table 7. All designed templates obtain higher performance than the random, but still at a low region in the second level compared with supervised learning. This result reveals that IDRR is still tricky for ChatGPT and cannot solve easily at current state, consistent with the result in Chan et al. (2023). The structural path template outperforms the other two templates, proving the help of the structural form for ChatGPT to understand this task. The F1 score of each second level is shown in Figure 10 in Appendix and illustrates the effectiveness to distinguish various second-level senses among the *Expansion* top class. More case examples and discussions refer to Appendix B.5.

Model	F1 (Top)	Acc (Top)	F1 (Second)	Acc (Second)
Random	23.44	32.18	6.48	8.78
ChatGPT$_{\text{label}}$	43.37	48.51	16.17	26.95
ChatGPT$_{\text{label \& con.}}$	43.99	49.28	17.55	30.32
ChatGPT$_{\text{structure}}$	44.09	50.24	19.88	31.95

Table 7: The performance of ChatGPT performs on the PDTB (*Ji*) test set. ChatGPT$_{label\&con.}$ means predicting the label and connective, and ChatGPT$_{structure}$ means adopting our structural path prompt template.

Model	Acc	F1
Pitler and Nenkova (2009)	94.15	-
Dai and Huang (2018)	94.46	93.70
Dai and Huang (2019)	95.39	94.84
Zhou et al. (2022)	94.78	93.59
Varia et al. (2019)	96.20	95.48
Fine-tuning (T5-large) w/o Connective	74.47	72.38
Fine-tuning (T5-large) w/ Gold Connective	95.41	94.94
DiscoPrompt (T5-large) w/ Connective Mask	78.35	74.62
DiscoPrompt (T5-large) w/ Gold Connective	**96.73**	**95.64**

Table 8: Explicit Top-level sense classification results on PDTB (*Ji*). "w/o Connective" and "w/ Connective Mask" regard the EDRR as IDRR.

5.6 Generalization to Explicit Discourse Relation Classification Task

To demonstrate the generalization ability of our model, we transfer and adapt our method to the explicit discourse relation recognition (EDRR) task. We simply replace the first [MASK] between two arguments with the gold connective for each instance in EDRR. Following the previous works (Varia et al., 2019; Zhou et al., 2022), the second-level class is the same as our implicit one setting. In Table 8, our model slightly outperforms previous SOTA models on the top-level sense prediction. DiscoPrompt consistently outperforms fine-tuning under different settings, and we observe a larger margin with absenting connectives.

6 Conclusion

In this paper, we introduce a path prediction method for tackling the IDRR task by utilizing the hierarchical structural information and prior knowledge of connectives. Combining label structures in natural language with prompt tuning successfully takes a step further in this task as well as other generalized settings, e.g., prompt adaptation and explicit relation detection. Our model achieves new SOTA performance on PDTB 2.0 and CoNLL-2016 data, and we hope our detailed discussions can help communities in discourse fields.

Limitations and Future Work

Limited Utilized Knowledge The main limitation of our method is the limited utilized knowledge. Since our prompt tuning-based method tests on Implicit Discourse Relation Recognition (IDRR) task, the elicited knowledge only comes from the dataset of this task and the model pre-training corpora. This constraint restricts the capability owing to the reporting bias (Gordon and Durme, 2013) in the pre-training models (PLMs). Moreover, the relatively few training data of several second-level classes resulting from the highly skewed label distribution problem requires extensive knowledge to make the model understand data instances and the task. Although we impose the prior human knowledge against the IDRR task from the input template designing to the discourse connectives selection, the knowledge source still only comes from our prior knowledge and the elicited knowledge of PLMs. As a result, even our method obtains a valid score in all second-level classes except the *Cont.Pragmatic Cause* displayed in Table 6, some second-level senses, which are the same as previous studies, cannot receive a satisfactory performance (e.g., *Comp.Concession* and *Expa.List*). The future work for this issue is to integrate more abundant knowledge and equip the model with more vital abilities. For example, grounding the arguments pair on the relevant nodes of the knowledge graph for each data instance (Lin et al., 2019) or knowledge distillation from large language models to provides more contextual information and enhances the capability of the model on this task.

Limited Predicted Connectives Another area for improvement is the prediction of extensive connectives. Although our model includes the preselected connectives as our third layer of a designed hierarchy tree, we do not include the ground truth of connectives as our third layer. Because including these extensive connectives to form many leaves will result in many paths (more than 100). This limitation may be addressed in future works by utilizing the pruning algorithms for reducing a lot of redundant nodes and leaves on each instance to enhance effectiveness and efficiency.

Ethics Statement

In this work, we conformed to recognized privacy practices and rigorously followed the data usage policy. We declare that all authors of this paper acknowledge the *ACM Code of Ethics* and honor the code of conduct. This paper presents a method to utilize the interaction information between different layers, inherent sense label structure, and prior knowledge of connectives in the implicit discourse recognition task. The PDTB 2.0 and CoNLL-2016 dataset were used to train and assess the ability of the pre-trained language model on this task. The PDTB2.0 and CoNLL2016-Test dataset is collected from the Wall Street Journal (WSJ) articles, while the CoNLL2016-Blind dataset is derived from newswire texts, the primary language is English based and belongs to the news domain. We can foresee no immediate social consequences or ethical issues as we do not introduce social/ethical bias into the model or amplify any bias from the data. Therefore, these two datasets are not required to perform further actions to check the offensive content.

Acknowledgements

The authors of this paper were supported by the NSFC Fund (U20B2053) from the NSFC of China, the RIF (R6020-19 and R6021-20) and the GRF (16211520 and 16205322) from RGC of Hong Kong, the MHKJFS (MHP/001/19) from ITC of Hong Kong and the National Key R&D Program of China (2019YFE0198200) with special thanks to HKMAAC and CUSBLT. We also thank the support from NVIDIA AI Technology Center (NVAITC) and the UGC Research Matching Grants (RMGS20EG01-D, RMGS20CR11, RMGS20CR12, RMGS20EG19, RMGS20EG21, RMGS23CR05, RMGS23EG08).

References

Hongxiao Bai and Hai Zhao. 2018. Deep enhanced representation for implicit discourse relation recognition. In *COLING*, pages 571–583.

Antoine Bosselut, Asli Celikyilmaz, Xiaodong He, Jianfeng Gao, Po-Sen Huang, and Yejin Choi. 2018. Discourse-aware neural rewards for coherent text generation. In *NAACL-HLT*, pages 173–184.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Chunkit Chan, Jiayang Cheng, Weiqi Wang, Yuxin Jiang, Tianqing Fang, Xin Liu, and Yangqiu Song. 2023. Chatgpt evaluation on sentence level relations: A focus on temporal, causal, and discourse relations. *CoRR*, abs/2304.14827.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Zeyu Dai and Ruihong Huang. 2018. Improving implicit discourse relation classification by modeling inter-dependencies of discourse units in a paragraph. In *NAACL-HLT*, pages 141–151.

Zeyu Dai and Ruihong Huang. 2019. A regularization approach for incorporating event knowledge and coreference relations into neural discourse parsing. In *EMNLP*, pages 2974–2985.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Haitao Zheng, and Maosong Sun. 2022. Openprompt: An open-source framework for prompt-learning. In *ACL*, pages 105–113.

Zujun Dou, Yu Hong, Yu Sun, and Guodong Zhou. 2021. CVAE-based re-anchoring for implicit discourse relation classification. In *Findings of EMNLP*, pages 1275–1283.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *ACL/IJCNLP*, pages 3816–3830.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *AKBC*, pages 25–30.

Fengyu Guo, Ruifang He, Jianwu Dang, and Jian Wang. 2020. Working memory-driven neural networks with a novel knowledge enhancement paradigm for implicit discourse relation recognition. In *AAAI*, pages 7822–7829.

Ruifang He, Jian Wang, Fengyu Guo, and Yugui Han. 2020. Transs-driven joint learning architecture for implicit discourse relation recognition. In *ACL*, pages 139–148.

Yangfeng Ji and Jacob Eisenstein. 2015. One vector is not enough: Entity-augmented distributed semantics for discourse relations. *Trans. Assoc. Comput. Linguistics*, 3:329–344.

Congcong Jiang, Tieyun Qian, and Bing Liu. 2022a. Knowledge distillation for discourse relation analysis. In *WWW*, pages 210–214.

Feng Jiang, Yaxin Fan, Xiaomin Chu, Peifeng Li, and Qiaoming Zhu. 2021. Not just classification: Recognizing implicit discourse relation on joint modeling of classification and generation. In *EMNLP*, pages 2418–2431.

Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. 2023. Lion: Adversarial distillation of closed-source large language model. *CoRR*, abs/2305.12870.

Yuxin Jiang, Linhan Zhang, and Wei Wang. 2022b. Global and local hierarchy-aware contrastive framework for implicit discourse relation recognition. *CoRR*, abs/2211.13873.

Najoung Kim, Song Feng, R. Chulaka Gunasekara, and Luis A. Lastras. 2020. Implicit discourse relation classification: We need to talk about evaluation. In *ACL*, pages 5404–5414.

Yudai Kishimoto, Yugo Murawaki, and Sadao Kurohashi. 2020. Adapting BERT to implicit discourse relation classification with a focus on discourse connectives. In *LREC*, pages 1152–1158.

Murathan Kurfali and Robert Östling. 2021. Let's be explicit about that: Distant supervision for implicit discourse relation classification via connective prediction. *CoRR*, abs/2106.03192.

Man Lan, Jianxiang Wang, Yuanbin Wu, Zheng-Yu Niu, and Haifeng Wang. 2017. Multi-task attention-based neural networks for implicit discourse relationship representation and identification. In *EMNLP*, pages 1299–1308.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, pages 3045–3059.

Jiaqi Li, Ming Liu, Bing Qin, and Ting Liu. 2022. A survey of discourse parsing. *Frontiers Comput. Sci.*, 16(5):165329.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL/IJCNLP*, pages 4582–4597.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *EMNLP-IJCNLP*, pages 2829–2839.

Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing implicit discourse relations in the penn discourse treebank. In *EMNLP*, pages 343–351.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated knowledge prompting for commonsense reasoning. In *ACL*, pages 3154–3169.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Xin Liu, Jiefu Ou, Yangqiu Song, and Xin Jiang. 2020. On the importance of word and sentence representation learning in implicit discourse relation classification. In *IJCAI*, pages 3830–3836.

Xin Liu, Jiefu Ou, Yangqiu Song, and Xin Jiang. 2021b. Exploring discourse structures for argument impact classification. In *ACL/IJCNLP*, pages 3958–3969.

Yang Liu and Sujian Li. 2016. Recognizing implicit discourse relations via repeated reading: Neural networks with multi-level attention. In *EMNLP*, pages 1224–1233.

Yang Liu, Sujian Li, Xiaodong Zhang, and Zhifang Sui. 2016. Implicit discourse relation classification via multi-task neural networks. In *AAAI*, pages 2750–2756.

Wanqiu Long and Bonnie Webber. 2022. Facilitating contrastive learning of discourse relational senses by exploiting the hierarchy of sense relations. In *EMNLP*, page 10704–10716.

Linh The Nguyen, Ngo Van Linh, Khoat Than, and Thien Huu Nguyen. 2019. Employing the correspondence of relations and connectives to identify implicit discourse relations via label embeddings. In *ACL*, pages 4201–4207.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

TB OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. *OpenAI*.

Emily Pitler and Ani Nenkova. 2009. Using syntax to disambiguate explicit discourse connectives in text. In *ACL*, pages 13–16.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K. Joshi, and Bonnie L. Webber. 2008. The penn discourse treebank 2.0. In *LREC*.

Lianhui Qin, Zhisong Zhang, and Hai Zhao. 2016a. Implicit discourse relation recognition with context-aware character-enhanced embeddings. In *COLING*, pages 1914–1924.

Lianhui Qin, Zhisong Zhang, and Hai Zhao. 2016b. A stacking gated neural architecture for implicit discourse relation classification. In *EMNLP*, pages 2263–2270.

Lianhui Qin, Zhisong Zhang, Hai Zhao, Zhiting Hu, and Eric P. Xing. 2017. Adversarial connective-exploiting networks for implicit discourse relation classification. In *ACL*, pages 1006–1017.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Attapol Rutherford and Nianwen Xue. 2015. Improving the inference of implicit discourse relations via classifying explicit discourse connectives. In *NAACL-HLT*.

Attapol Rutherford and Nianwen Xue. 2016. Robust non-explicit neural discourse parser in english and chinese. In *SIGNLL*, pages 55–59.

Teven Le Scao and Alexander M. Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2627–2636. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *EACL*, pages 255–269.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *ICML*, pages 4603–4611.

Wei Shi and Vera Demberg. 2019a. Learning to explicitate connectives with seq2seq network for implicit discourse relation classification. In *IWCS*, pages 188–199.

Wei Shi and Vera Demberg. 2019b. Next sentence prediction helps implicit discourse relation classification within and across domains. In *EMNLP-IJCNLP*, pages 5789–5795.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Siddharth Varia, Christopher Hidey, and Tuhin Chakrabarty. 2019. Discourse relation prediction: Revisiting word pairs with convolutional networks. In *SIGDIAL*, pages 442–452.

Han Wang, Canwen Xu, and Julian J. McAuley. 2022a. Automatic multi-label prompting: Simple and interpretable few-shot classification. In *NAACL-HLT*, pages 5483–5492.

Jianxiang Wang and Man Lan. 2016. Two end-to-end shallow discourse parsers for english and chinese in conll-2016 shared task. In *SIGNLL*, pages 33–40.

Zihan Wang, Peiyi Wang, Tianyu Liu, Yunbo Cao, Zhifang Sui, and Houfeng Wang. 2022b. HPT: hierarchy-aware prompt tuning for hierarchical text classification. *CoRR*, abs/2204.13413.

Changxing Wu, Liuwen Cao, Yubin Ge, Yang Liu, Min Zhang, and Jinsong Su. 2022. A label dependence-aware sequence generation model for multi-level implicit discourse relation recognition. In *AAAI*, pages 11486–11494.

Changxing Wu, Xiaodong Shi, Yidong Chen, Jinsong Su, and Boli Wang. 2017. Improving implicit discourse relation recognition with discourse-specific word embeddings. In *ACL*, pages 269–274.

Yang Xu, Yu Hong, Huibin Ruan, Jianmin Yao, Min Zhang, and Guodong Zhou. 2018. Using active learning to expand training data for implicit discourse relation recognition. In *EMNLP*, pages 725–731.

Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Attapol Rutherford, Bonnie L. Webber, Chuan Wang, and Hongmin Wang. 2016. Conll 2016 shared task on multilingual shallow discourse parsing. In *SIGNLL*, pages 1–19.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, pages 5754–5764.

Wanjun Zhong, Yifan Gao, Ning Ding, Yujia Qin, Zhiyuan Liu, Ming Zhou, Jiahai Wang, Jian Yin, and Nan Duan. 2022. Proqa: Structural prompt-based pre-training for unified question answering. In *NAACL-HLT*, pages 4230–4243.

Hao Zhou, Man Lan, Yuanbin Wu, Yuefeng Chen, and Meirong Ma. 2022. Prompt-based connective prediction method for fine-grained implicit discourse relation recognition. *CoRR*, abs/2210.07032.

A Appendix for Experimental Settings

A.1 DataSet

The Penn Discourse Treebank 2.0 (PDTB 2.0) PDTB 2.0[3] is a large-scale corpus containing 2,312 Wall Street Journal (WSJ) articles (Prasad et al., 2008), that employs a lexically-grounded approach to annotating discourse relations. This corpus includes three sense levels (i.e., classes, types, and sub-types) and naturally forms the sense hierarchy. In this dataset, we validate our model on two popular settings of the PDTB 2.0 dataset, which are the Ji-setting (Ji and Eisenstein, 2015) and Lin-setting (Lin et al., 2009). The former one following Ji and Eisenstein (2015) to split sections 2-20, 0-1, and 21-22 as training, validation, and test sets respectively, while the latter follows Lin et al. (2009) split sections 2-21, 22, 23 as training, validation, and test sets respectively. We evaluate our model on the four top-level implicit discourse relations and the 11 major second-level implicit discourse senses by following previous works (Wu et al., 2022; Long and Webber, 2022; Zhou et al., 2022). The data statistics of the top-level and second-level senses are displayed in Table 9 and Table 10.

The CoNLL-2016 Shared Task (CoNLL16) The CoNLL-2016 shared task[4] provides more abundant annotation (e.g., second-level sense type) for shadow discourse parsing. This task includes two test sets, the PDTB section 23 (CoNLL-Test) and newswire texts (CoNLL-Blind), that comply with the PDTB annotation guidelines. Compared with PDTB 2.0, CoNLL16 includes more new class sense (e.g., *Contingency.Condition*) and merges several labels to annotate new labels. For example, *Contingency.Pragmatic cause* is merged into *Contingency.Cause.Reason* to remove the former type with very few samples. In this paper, we follow Wang and Lan (2016); Lan et al. (2017); Liu et al. (2020) to perform the experiments on this CoNLL-2016 dataset and validate the performance of our model in the top- and second-level sense.

[3]The License of the PDTB 2.0 dataset is LDC User Agreement for Non-Members, and this paper is consistent with their intended use for research purposes. This dataset download from https://catalog.ldc.upenn.edu/LDC2008T05.

[4]CoNLL16 dataset download from https://www.cs.brandeis.edu/~clp/conll16st/dataset.html.

Top-level Senses	Train	Val.	Test
Comparison	1,942	197	152
Contingency	3,342	295	279
Expansion	7,004	671	574
Temporal	760	64	85
Total	12,362	1,183	1,046

Table 9: Statistics of four top-level implicit senses in PDTB 2.0.

Second-level Senses	Train	Val.	Test
Comp.Concession	180	15	17
Comp.Contrast	1566	166	128
Cont.Cause	3227	281	269
Cont.Pragmatic cause	51	6	7
Exp.Alternative	146	10	9
Exp.Conjunction	2805	258	200
Exp.Instantiation	1061	106	118
Exp.List	330	9	12
Exp.Restatement	2376	260	211
Temp.Asynchronous	517	46	54
Temp.Synchrony	147	8	14
Total	12406	1165	1039

Table 10: The implicit discourse relation data statistics of second-level types in PDTB 2.0.

A.2 DiscoPrompt Implementation Details

DiscoPrompt is prompt tuning upon T5-model, and we also validate our method over various model scales, including T5-base, T5-large, and T5-11b. Figure 12 shows the heat map of highly frequent connectives on CoNLL2016, and the label words are in Table 12. Generally, the overall configuration follows the setting in Lester et al. (2021) and sets the learnable prompt length as 20. The training was implemented using cross-entropy loss with 30,000 training steps, which selects the model that yields the best performance on the validation set. We adopt an Adafactor (Shazeer and Stern, 2018) optimizer with various learning rate ranges for different dataset settings. The batch size and maximum input sequence are 4 and 350, respectively. The maximum generates sequence length of the encoder is 10. Our model is conducted on two 32GB NVIDIA V100 GPUs, except for the T5-11b scale on two 48GB NVIDIA A6000 GPUs. The running time for T5-base is around 8 hours, while T5-large is about 19 hours.

Since we are interested in the ability of our method to adopt a larger-scale model on this task, we tested the T5-11b model on various datasets. Most of the configuration is the same as the above T5-large version. The slight differences in hyperparameters are batch size is one and gradient accumulation step is 16. The running time of the T5-11b model is around 50 hours. The tailored prompt template is shown in Figure 8. The specific hyperparameters of implementation details for DiscoPrompt (T5-large) and DiscoPrompt (T5-11b) are displayed in Table 11. The frozen pre-train T5 model download from *HuggingFace*, and our model inheritance and modification from *OpenPrompt* (Ding et al., 2022).

Dataset	Hyperparameters
PDTB (*Ji*)	LR space: {9e-2, 9e-1}, LR*: 3e-1, BS: 4, gradient accumulation step:1
PDTB (*Lin*)	LR space: {9e-4, 9e-3}, LR*: 2e-4, BS: 4, gradient accumulation step:1
CoNLL16 (*Test*)	LR space: {9e-2, 9e-1}, LR*: 9e-2, BS: 4, gradient accumulation step:1
CoNLL16 (*Blind*)	LR space: {9e-2, 9e-1}, LR*: 9e-2, BS: 4, gradient accumulation step:1
PDTB (*Ji*)	LR space: {9e-4, 9e-3}, LR*: 4e-4, BS: 1, gradient accumulation step:16
PDTB (*Lin*)	LR space: {9e-4, 9e-3}, LR*: 5e-4, BS: 1, gradient accumulation step:16
CoNLL16 (*Test*)	LR space: {9e-5, 9e-4}, LR*: 9e-5 BS: 1, gradient accumulation step:16
CoNLL16 (*Blind*)	LR space: {9e-4, 9e-3}, LR*: 2e-4, BS: 1, gradient accumulation step:16

Table 11: The hyperparameters of implementation details for DiscoPrompt (T5-large) and DiscoPrompt (T5-11b). The upper part is for the T5-large version in four datasets, while the bottom is for the T5-11b version. "LR space", "LR*", and "BS" refer to learning rate searching space, optimal learning rate, and batch size, respectively.

Top-level	Second-level	Connectives
Comparison	Concession Contrast	nonetheless but
Contingency	Reason Result Condition	because so if
Expansion	Alternative Chosen Conjunction Instantiation Exception Restatement	unless instead and for example except indeed
Temporal	Precedence Succession Synchrony	before previously when

Table 12: The label word set on CoNLL2016 dataset.

A.3 Baseline Models

To exhibit the effectiveness of our proposed method, we compared it with previous works on the PDTB 2.0 and CoNLL16 datasets. In this section, we mainly describe some recently published

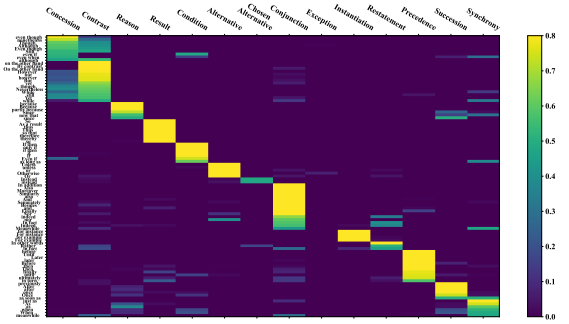

Figure 5: Prior probabilities of CoNLL16 frequent connectives.

baselines, and more baselines can be found in Table 14.

Common Baselines for PDTB 2.0 and CoNLL16:

- **RWP-CNN** (Varia et al., 2019): a convolutional neural networks-based method to model word pairs in the arguments in a discourse relation.
- **BMGF-RoBERTa** (Liu et al., 2020): a RoBERTa-based model, which contains a robust contextualized representation module, a bilateral matching module to capture the interaction between arguments, and a global information fusion module to derive final representations for labels.
- **XLNet** (Kim et al., 2020): it fine-tunes XLNet model (Yang et al., 2019) for IDRR task to predict the flat label in each layer of discourse relation sense.
- **T5 (Fine-Tuning)** (Raffel et al., 2020): Fine-tune a T5-model based on specifics tailored input text in various settings with a comparison of our model. The Implementation details are described in Appendix A.5.
- **Prefix-Tuning (T5)** (Li and Liang, 2021): a lightweight method concatenates the tunable prefix tokens before the discrete input text, keeps language model parameters frozen, and optimizes these continuous task-specific prefix tokens. The implementation details of the Prefix-Tuning methods are appended in Appendix A.4.
- **Prompt-Tuning (T5)** (Lester et al., 2021): a vanilla Prompt Tuning-based model conditioning on a frozen model, releasing the constraints of the prompt templates from discrete to learnable prompts. The implementation details of the prompt tuning methods are appended in Appendix A.4.
- **PCP** (Zhou et al., 2022): a prompt-based connective prediction method for IDRR by adopting the RoBERTa model. This method utilizes the strong correlation between connectives and discourse relations to map the predicted connectives to respective implicit discourse relations.

Baselines for PDTB 2.0:

- **DER** (Bai and Zhao, 2018): a model enhanced with multiple grained text representations, including character, subword, word, sentence, and sentence pair levels.
- **MTL-MLoss** (Nguyen et al., 2019): a multi-task learning neural model that predicts labels and connectives simultaneously by leveraging the dependence between them.
- **ELMo-C&E** (Dai and Huang, 2019): a neural model that employs a regularization approach to utilize the external event knowledge and coreference relations.
- **TransS** (He et al., 2020): a TransS-driven joint learning model which translates the discourse relations in low-dimensional embedding space (i.e., TransS), and simultaneously learns the semantic features of arguments.
- **CG-T5** (Jiang et al., 2021): a joint model that recognizes the relation label and generates the desired target sentence containing the meaning of relations simultaneously.
- **OTMT(XLNet)** (Jiang et al., 2022a): an XLNet (Yang et al., 2019) based model exploits the knowledge distillation (KD) technique for discourse relation recognition task.
- **LDSGM** (Wu et al., 2022): a label dependence-aware sequence generation model that integrates the global representation of an input instance, level-specific contexts, and the label dependence decoded by graph convolutional network (GCN) to obtain better label embeddings, and then employ the label sequence decoder to output the predicted labels.
- **GOLF** (Jiang et al., 2022b): a global and local hierarchy-aware contrastive framework, to model and capture the information from these two kinds of hierarchies with the aid of contrastive learning.

- **ContrastiveIDRR** (Long and Webber, 2022): a contrastive learning method for incorporating the sense hierarchy into the recognition process and using the hierarchy to select the negative examples.

Baselines for CoNLL16:

- **CoNLL Baseline** (Rutherford and Xue, 2016): a neural classifier requires word vectors and a simple feed-forward training procedure.
- **MTL-Attn-LSTM** (Lan et al., 2017): a multi-task attention-based LSTM neural network model that exploits explicit discourse relations in PDTB and unannotated external data in a multi-task joint learning framework.

A.4 Implementation Details of the Prefix-Tuning and Prompt Tuning

In our paper, we implement the prefix tuning (Li and Liang, 2021) and prompt tuning (Lester et al., 2021) methods as the baselines for comparison with our model. We proposed several templates for searching for their best performance in these two methods. The experimental details for these two methods include the template and hyperparameter search. Moreover, there are 154 tokens, including textual tokens (non-tunable tokens) and tunable tokens, in our prompt template. For a fair comparison, we insert 154 tunable tokens into the respective prompt template in these two baselines.

Prefix-Tuning Following the setting of prefix tuning (Li and Liang, 2021), we implemented several designed templates on the PDTB 2.0 JI setting and the templates shown in figure 6. In these templates, we find that the **prefix-prompt template three** is better among all templates, and we adopted this template for further comparison with our method. The overall configuration of this model follows the settings of prefix tuning (Li and Liang, 2021). The batch size and maximum sequence length of this model are 8 and 350. The training is performed using cross-entropy loss with an Adafactor optimizer (Shazeer and Stern, 2018) and a learning rate selecting in 0.3, 0.5, 0.8 yields the best performance on the validation set, and the training steps are 30,000.

Prompt-Tuning For the prompt tuning method, we implemented several designed templates on the PDTB 2.0 JI setting and the templates shown in figure 7. In these templates, we find that the **prompt tuning template two** is better among all templates, and adopted this template for further comparison with our method. The overall configuration of this model follows the settings of prefix tuning (Lester et al., 2021). The batch size and maximum sequence length of this model are 8 and 350. The training is performed using cross-entropy loss with an Adafactor optimizer (Shazeer and Stern, 2018) and a learning rate selecting in 0.3, 0.5, 0.8 yields the best performance on the validation set, and the training steps are 30,000.

A.5 Implementation Details of T5 Model Fine-Tuning

Here we provide the fine-tuning details for T5 base and large models on various datasets.

Model Input and Output In main experiments, T5-model fine-tuning as the competitive baseline, we concatenate two arguments with an "</s>" at the end of the sequence as input. The T5 model asked to generate the top-level labels, and the second-level labels with concatenating by commas (e.g., Comparison.Contrast) given the data input. For the experiments to test the transferred template on the fine-tuning paradigm, the "T5-adapt" model in section 5.4 concatenate the hierarchy tree prompt in Figure 8 before the two arguments as input. Then we concatenate a prompt message "The path is " before the original output. Furthermore, for the setting "T5-large (fine-tune) (w/ connective)" in the EDRR task (Section 5.6), it required inserting the connectives between two arguments. Therefore, we use the text span named "FullRawText" in the dataset with an additional "</s>" at the end as input.

Hyperparameter Search We first conduct a preliminary experiment to determine the range of hyper-parameters. Then, we search for the learning rate within $\{3e-4, 1e-4\}$ and warmup steps within $\{0, 100\}$. For the T5-base model, we set the training batch size as 8, and the model is evaluated with a batch size of 128 every 150 steps. For the T5-large model, the training and evaluation batch sizes are set as 16 and 64, respectively. The model is optimized with an AdamW optimizer with a linear learning rate schedule. The test performance of the model with the best validation accuracy is reported.

Model	Parameters
BMGF-RoBERTa (Liu et al., 2020)	2.3M
XLNet(base, cased) (Kim et al., 2020)	110M
XLNet(large, cased) (Kim et al., 2020)	340M
OTMT(XLNet-base) (Jiang et al., 2022a)	110M
OTMT(XLNet-large) (Jiang et al., 2022a)	340M
Fine-Tuning (T5-base) (Raffel et al., 2020)	220M
Fine-Tuning (T5-large) (Raffel et al., 2020)	770M
Prefix-Tuning (T5-base) (Li and Liang, 2021)	0.12M
Prefix-Tuning (T5-large) (Li and Liang, 2021)	0.16M
Prompt-Tuning (T5-base) (Lester et al., 2021)	0.12M
Prompt-Tuning (T5-large) (Lester et al., 2021)	0.16M
LDSGM (Wu et al., 2022)	128M
ContrastiveIDRR (Long and Webber, 2022)	125M
PCP(RoBERTa-base) (Zhou et al., 2022)	124M
PCP(RoBERTa-large) (Zhou et al., 2022)	335M
DiscoPrompt (T5-base)	1.2M
DiscoPrompt (T5-large)	2.1M

Table 13: The approximation of learnable parameters for models. "M" stands for million learnable parameters.

A.6 The Approximation of Learnable Parameters

To show the efficiency of our method, we append the approximation of learnable parameters for all models, including our model and baselines. The approximation of learnable parameters is listed in Table 13.

B Appendix for Evaluation Result and Analysis

B.1 Performance of Baselines in PDTB 2.0

In this section, we list extensive baselines in Table 14 for comparison with our method.

B.2 Ablation study on the DiscoPrompt

Prompt Template Searching We perform the prompt template research on our designed prompt, and all prompt searching templates are listed in Figure 8, and the performance is shown in Table 15. Our finalized optimal template inserts the connectives between two arguments to improve the textual coherence of input context and results in the PLMs easy to understand input. Therefore, this template performs better than other designed templates.

Continuous Prompt Length The continuous prompt (i.e., learnable prompt tokens) length is another factor that influences the performance of our model. Hence, we implement various prompt lengths of 10, 20, 50, and 100. The performance is in Table 16, and the optimal continuous prompt length is 20, which provides the best performance among all the prompt lengths and is the default prompt length for implementing other experiments. Adopting more prompt length than 20 on our method will not significantly increase this task's performance on various evaluation metrics.

B.3 Performance of Label-wise F1 Score on Top and Second level

The performance (F1 score%) of more baselines for comparison with our model in Top-level and Second-level shown in Table 17 and Table 19.

B.4 Performance of Designed Prompt For T5 Fine-Tuning

The performance comparison of the T5-large fine-tuning with and without using our designed template on the PDTB 2.0 is displayed in Figure 9. The detailed experimental result for PDTB 2.0 and CoNLL16 dataset is shown in Table 18.

B.5 Discussion and Case Example for ChatGPT

With the powerful ability of large language model exhibited on numerous tasks (OpenAI, 2022, 2023; Taori et al., 2023; Chiang et al., 2023; Jiang et al., 2023), we are curious about the capability of ChatGPT on zero-shot IDRR task. Hence, we test the ability of ChatGPT [6] with three designed templates on the PDTB (*Ji*) test set. These templates include: 1) predict the class label only, 2) predict the class label with connectives, and 3) predict the class label with connectives in a structural path form. Moreover, the input template with in-context learning highly relies on the training examples selected as the prefix instruction part of the prompt template. The performance of this model is high variance with the chosen examples vary. Therefore, this template is not taken into account in this section. The performance of the random guess model is obtained by averaging the performance of 5 runs. A prediction is regarded as wrong if ChatGPT generates the answer out of the range of label words. An interesting finding is that the ChatGPT with label-only template tends to predict many temporally related instances to the *Contingency.Cause* second-level sense result in poor performance on *Temporal.synchrony* second-level sense shown in Figure 10. The input template and two case examples are shown in Table 20 and Table 21.

[5] We use their model without a data augmentation version for a fair comparison in Table 2. This model with the data augmentation version is also appended in this table.

[6] The demonstration and details of ChatGPT are on the website https://openai.com/blog/chatgpt/

Models	Ji (Top)		Ji (Sec)		Lin (Top)		Lin (Sec)	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc
Lin et al. (2009)	-	-	-	-	-	-	-	40.20
Ji and Eisenstein (2015)	-	-	-	44.59	-	-	-	-
Liu et al. (2016)	44.98	57.27	-	-	-	-	-	-
Qin et al. (2016a)	-	-	-	45.04	-	-	-	43.81
Liu and Li (2016)	46.29	57.57	-	-	-	-	-	-
Wu et al. (2017)	44.84	58.85	-	-	-	-	-	-
Lan et al. (2017)	47.80	57.39	-	-	-	-	-	-
Qin et al. (2017)	-	-	-	46.23	-	-	-	44.65
Xu et al. (2018)	44.48	60.63	-	-	-	-	-	-
Dai and Huang (2018)	48.82	57.44	-	-	-	-	-	-
Bai and Zhao (2018)	51.06	-	-	48.22	-	-	-	45.73
Shi and Demberg (2019a)	46.40	61.42	-	47.83	-	-	-	45.82
Varia et al. (2019)	50.20	59.13	-	-	-	-	-	-
Dai and Huang (2019)	52.89	59.66	-	48.23	-	-	-	-
Nguyen et al. (2019)	53.00	-	-	49.95	-	-	-	46.48
Shi and Demberg (2019b)	-	-	-	53.23	-	-	-	-
He et al. (2020)	-	-	-	-	51.24	59.94	-	-
Guo et al. (2020)	47.90	57.25	-	-	-	-	-	-
Kishimoto et al. (2020)	58.48	65.26	-	54.32	-	-	-	-
Liu et al. (2020)	63.39	69.06	35.25	58.13	58.54	68.66	39.15	53.96
Jiang et al. (2021)	57.18	-	37.76	-	-	-	-	-
Kurfali and Östling (2021)	59.24	-	39.33	55.42	-	-	-	-
Dou et al. (2021)	65.06	70.17	-	-	-	-	-	-
Wu et al. (2022)	63.73	71.18	40.49	60.33	-	-	-	-
Jiang et al. (2022b)	65.76	72.52	41.74	61.16	-	-	-	-
Long and Webber (2022)(w/o data augm.)[5]	67.85	71.70	45.54	59.19	-	-	-	-
Long and Webber (2022) (w data augm.)	69.60	72.18	49.66	61.69	-	-	-	-
BERT-base (Devlin et al., 2019)	43.17	62.14	26.32	50.24	43.44	63.46	26.70	49.87
BERT-large (Devlin et al., 2019)	57.06	67.59	30.02	54.57	56.06	68.40	38.68	56.53
XLNet(base, cased) (Kim et al., 2020)	59.33	66.35	36.36	54.73	56.16	68.05	36.23	55.82
XLNet (large, cased) (Kim et al., 2020)	63.58	69.52	38.24	61.29	58.97	72.17	40.71	58.77
OTMT (XLNet-base) (Jiang et al., 2022a)	60.78	68.89	-	56.65	-	-	-	56.37
OTMT (XLNet-large) (Jiang et al., 2022a)	64.46	72.34	-	61.06	-	-	-	61.62
Fine-Tuning (T5-base) (Raffel et al., 2020)	57.61	65.39	33.96	55.53	50.50	63.59	36.49	51.96
Fine-Tuning (T5-large) (Raffel et al., 2020)	61.37	69.69	38.04	57.65	58.12	71.13	42.04	59.40
Prefix-Tuning (T5-base) (Li and Liang, 2021)	25.87	52.45	7.49	31.09	25.08	54.18	8.45	26.37
Prefix-Tuning (T5-large) (Li and Liang, 2021)	63.74	71.51	39.73	59.77	58.06	69.84	36.86	56.53
Prompt-Tuning (T5-base) (Lester et al., 2021)	30.17	56.11	15.01	38.21	25.26	55.09	8.97	27.68
Prompt-Tuning (T5-large) (Lester et al., 2021)	66.95	71.99	44.08	60.15	59.92	71.02	40.75	60.44
PCP w/ RoBERTa-base (Zhou et al., 2022)	64.95	70.84	41.55	60.54	53.00	66.58	41.19	56.14
PCP w/ RoBERTa-large (Zhou et al., 2022)	67.79	73.80	44.04	61.41	52.75	71.13	43.04	60.44
DiscoPrompt (T5-base)	65.79	71.70	43.68	61.02	64.90	71.28	41.82	59.27
DiscoPrompt (T5-large)	**70.84**	**75.65**	**49.03**	**64.58**	**67.06**	**73.76**	**45.25**	**63.05**
DiscoPrompt (T5-11b)	75.34	78.06	52.42	68.14	72.78	77.55	47.18	67.62

Table 14: The accuracy (%) and F1 score (%) are evaluated on the PDTB 2.0 dataset.

Prefix-Tuning	Templates
Templates 1	[20 Continuous Prompt] [Argument 1] [Argument 2] [mask]
Templates 2	[154 Continuous Prompt] [Argument 1] [Argument 2] [mask]
Templates 3	[150 Continuous Prompt] [Argument 1] [Argument 2] The relation is [mask]

Figure 6: Prefix-Tuning Template Searching

Prompt Tuning	Templates
Templates 1	[150 Continuous Prompt] [Argument 1] [Argument 2] [4 Continuous Prompt] [mask]
Templates 2	[52 Continuous Prompt] [Argument 1] [51 Continuous Prompt] [Argument 2] [51 Continuous Prompt] [mask]

Figure 7: Prompt Tuning Template Searching

DiscoPrompt	Templates
Optimal Templates	[20 Continuous Prompt] Comparison -> Concession -> if ; Comparison -> Contrast -> however ; Contingency -> Cause -> so ; Contingency -> Pragmatic -> indeed; Expansion -> Alternative -> instead ; Expansion -> Conjunction -> also ; Expansion -> Instantiation -> for example; Expansion -> List -> and ; Expansion -> Restatement -> specifically ; Temporal -> Asynchronous -> before ; Temporal -> Synchrony -> when . [Argument 1] [mask] [Argument 2] The path is [mask] -> [mask];
Templates 1	[20 Continuous Prompt] Comparison -> Concession -> if ; Comparison -> Contrast -> however ; Contingency -> Cause -> so ; Contingency -> Pragmatic -> indeed; Expansion -> Alternative -> instead ; Expansion -> Conjunction -> also ; Expansion -> Instantiation -> for example; Expansion -> List -> and ; Expansion -> Restatement -> specifically ; Temporal -> Asynchronous -> before ; Temporal -> Synchrony -> when . [Argument 1] [Argument 2] The path is [mask] -> [mask] -> [mask]
Templates 2	[20 Continuous Prompt] Comparison -> Concession -> if ; Comparison -> Contrast -> however ; Contingency -> Cause -> so ; Contingency -> Pragmatic -> indeed; Expansion -> Alternative -> instead ; Expansion -> Conjunction -> also ; Expansion -> Instantiation -> for example; Expansion -> List -> and ; Expansion -> Restatement -> specifically ; Temporal -> Asynchronous -> before ; Temporal -> Synchrony -> when . [Argument 1] [mask] [Argument 2] The relation is [mask].[mask]
Templates 3	[20 Continuous Prompt] Comparison -> Concession -> if ; Comparison -> Contrast -> however ; Contingency -> Cause -> so ; Contingency -> Pragmatic -> indeed; Expansion -> Alternative -> instead ; Expansion -> Conjunction -> also ; Expansion -> Instantiation -> for example; Expansion -> List -> and ; Expansion -> Restatement -> specifically ; Temporal -> Asynchronous -> before ; Temporal -> Synchrony -> when . [Argument 1] [mask] [Argument 2] The relation is [mask] -> [mask]
Templates 4	[20 Continuous Prompt] Comparison -> Concession -> if ; Comparison -> Contrast -> however ; Contingency -> Cause -> so ; Contingency -> Pragmatic -> indeed; Expansion -> Alternative -> instead ; Expansion -> Conjunction -> also ; Expansion -> Instantiation -> for example; Expansion -> List -> and ; Expansion -> Restatement -> specifically ; Temporal -> Asynchronous -> before ; Temporal -> Synchrony -> when . [Argument 1] [Argument 2] The relation is [mask] -> [mask] -> [mask];
Templates 5	[20 Continuous Prompt] Comparison -> Concession -> if ; Comparison -> Contrast -> however ; Contingency -> Cause -> so ; Contingency -> Pragmatic -> indeed; Expansion -> Alternative -> instead ; Expansion -> Conjunction -> also ; Expansion -> Instantiation -> for example; Expansion -> List -> and ; Expansion -> Restatement -> specifically ; Temporal -> Asynchronous -> before ; Temporal -> Synchrony -> when . [Argument 1] [Argument 2] The relation is [mask] -> [mask]. The connective is [mask].

Figure 8: DiscoPrompt Template Searching. The "Optimal Templates" is the finalized optimal template for implementing experiments to compare with extensive baselines.

Model	F1 (Top)	Acc (Top)	F1 (Second)	Acc (Second)
DiscoPrompt (Optimal Template : The path is mask -> mask;)	**70.84**	**75.65**	**49.03**	**64.58**
DiscoPrompt (Template 1: The path is mask -> mask -> mask)	69.22	73.44	43.52	63.33
DiscoPrompt (Template 2: The relation is mask.mask)	67.55	74.21	44.81	64.20
DiscoPrompt (Template 3: The relation is mask -> mask)	69.70	74.01	48.61	64.10
DiscoPrompt (Template 4: The relation is mask -> mask -> mask;)	68.07	72.76	45.91	62.56
DiscoPrompt (Template 5: The relation is mask -> mask.The connective is mask .)	62.71	70.74	40.19	58.81

Table 15: Performance of various templates of our method with adopting T5-large model in PDTB (*Ji*) dataset. The details of various templates are shown in Figure 8.

Model	F1 (Top)	Acc (Top)	F1 (Second)	Acc (Second)
DiscoPrompt (T5-large)	**70.84**	**75.65**	**49.03**	**64.58**
Continuous Prompt Length (10)	67.17	72.47	43.56	62.66
Continuous Prompt Length (50)	69.64	74.40	45.06	63.91
Continuous Prompt Length (100)	68.39	73.92	42.77	64.20

Table 16: Performance of various continuous prompt lengths in our method DiscoPrompt (T5-large) on PDTB (*Ji*) dataset. The default continuous prompt length of our model is 20.

Model	Comp.	Cont.	Exp.	Temp.
Ji and Eisenstein (2015)	35.93	52.78	-	27.63
Rutherford and Xue (2015)	41.00	53.80	69.40	33.30
Liu et al. (2016)	37.91	55.88	69.97	37.17
Liu and Li (2016)[2]	39.86	54.48	70.43	38.84
Qin et al. (2016b)	38.67	54.91	71.50	32.76
Lan et al. (2017)	40.73	58.96	72.47	38.50
Bai and Zhao (2018)	47.85	54.47	70.60	36.87
Dai and Huang (2018)	46.79	57.09	70.41	45.61
Varia et al. (2019)	44.10	56.02	72.11	44.41
Nguyen et al. (2019)	48.44	56.84	73.66	38.60
Guo et al. (2020)	43.92	57.67	73.45	36.33
Liu et al. (2020)	59.44	60.98	77.66	50.26
Jiang et al. (2021)	55.40	57.04	74.76	41.54
Dou et al. (2021)	55.72	63.39	80.34	44.01
Long and Webber (2022)	65.84	63.55	79.17	69.86
DiscoPrompt (T5-base)	62.55	64.45	78.77	57.41
DiscoPrompt (T5-large)	67.13	69.76	81.61	64.86
DiscoPrompt (T5-11b)	**74.35**	**72.44**	**82.57**	**72.00**

Table 17: The performance for top-level classes on PDTB (*Ji*) in terms of F1 (%) (top-level multi-class classification).

Model(DataSet Settings)	Acc (Second)	F1 (Second)	Acc (Top)	F1 (Top)
T5 (PDTB (*Ji*))	57.65	38.04	69.69	61.37
T5-adapt(PDTB (*Ji*))	59.77	38.08	70.17	60.89
T5 (PDTB (*Lin*))	59.40	42.04	71.13	58.12
T5-adapt(PDTB (*Lin*))	59.53	42.83	71.91	61.03
T5 (CoNLL-Test)	58.88	34.66	70.87	58.74
T5-adapt(CoNLL-Test)	59.66	37.49	71.52	60.78
T5 (CoNLL-Blind)	54.3	24.63	73.07	56.28
T5-adapt(CoNLL-Blind)	56.07	26.85	74.61	57.77

Table 18: The performance comparison of the T5-large fine-tuning with and without using our designed template on the PDTB 2.0 and CoNLL16 dataset. "T5-adapt" means adopting our designed template in the fine-tuning process. Acc and F1 inside the brackets indicate the accuracy and F1 score.

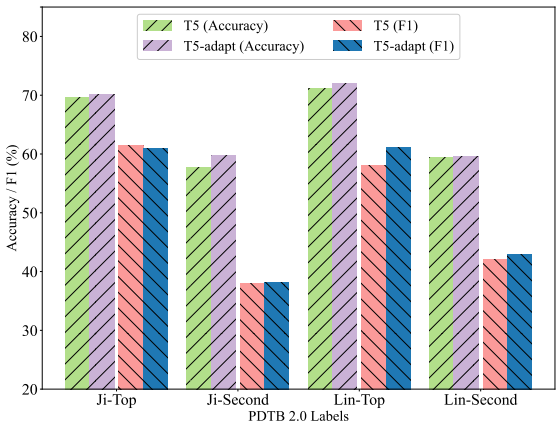

Figure 9: The performance comparison of the T5-large fine-tuning with and without using our designed template on the PDTB 2.0 dataset. "T5-adapt" means adopting our designed template in the fine-tuning process. Acc and F1 inside the brackets indicate the accuracy and F1 score.

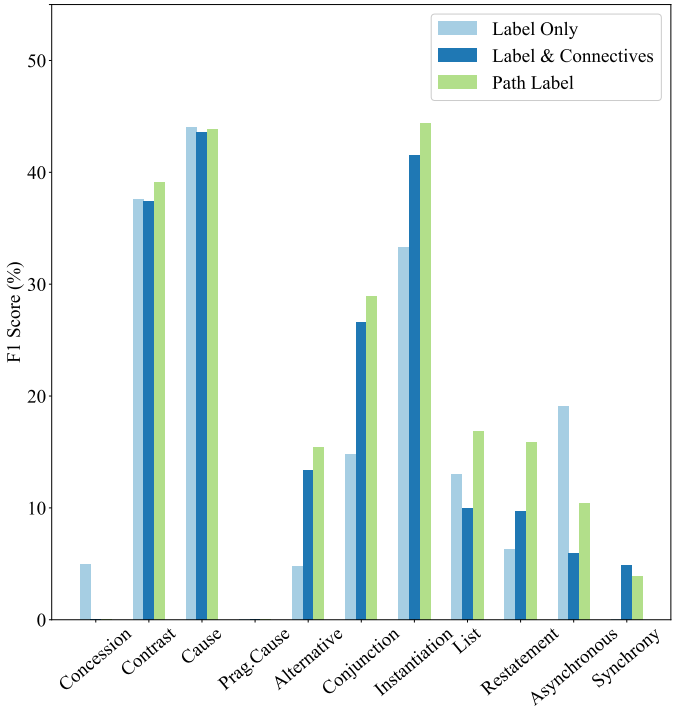

Figure 10: The performance comparison of various input prompt templates for ChatGPT. "Prag.Cause" stand for *Pragmatic cause* second level sense.

Second-level Label	**BMGF**	**LDSGM**	**PCP**	**ContrastiveIDRR**	**Ours$_{(base)}$**	**Ours$_{(large)}$**	**Ours$_{(11B)}$**
Temp.Asynchronous	56.18	56.47	57.81	59.79	57.69	64.15	**72.27**
Temp.Synchrony	0.0	0.0	0.0	**78.26**	0.0	50.00	33.33
Cont.Cause	59.60	64.36	65.64	65.58	63.83	69.66	**72.28**
Cont.PragmaticCause	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Comp.Contrast	59.75	63.52	63.88	62.63	59.26	62.88	**70.63**
Comp.Concession	0.0	0.0	8.00	0.0	**9.09**	**9.09**	0.0
Expa.Conjunction	60.17	57.91	57.78	58.35	61.08	60.09	**62.84**
Expa.Instantiation	67.96	72.60	74.01	73.04	69.96	74.17	**76.60**
Expa.Restatement	53.83	58.06	61.00	60.00	58.45	65.24	**65.98**
Expa.Alternative	60.00	63.46	66.67	53.85	72.73	60.00	**84.21**
Expa.List	0.0	8.98	29.63	34.78	37.50	24.00	**38.46**

Table 19: The label-wise F1 scores for the second-level labels on PDTB (*Ji*) (second-level multi-class classification).

Query Template: Argument 1: Right away you notice the following things about a Philip Glass concert. Argument 2: It attracts people with funny hair. What is the relation label between Argument 1 and Argument 2? Select from the candidates.
1. Comparison.Concession
2. Comparison.Contrast
3. Contingency.Cause
4. Contingency.Pragmatic
5. Expansion.Alternative
6. Expansion.Conjunction
7. Expansion.Instantiation
8. Expansion.List
9. Expansion.Restatement
10. Temporal.Asynchronous
11. Temporal.Synchrony
ChatGPT: Expansion.Instantiation

Query Template: Argument 1: Right away you notice the following things about a Philip Glass concert. Argument 2: It attracts people with funny hair. What is the relation and connective between Argument 1 and Argument 2? Select from the candidates.
1. Comparison.Concession, if
2. Comparison.Contrast, however
3. Contingency.Cause, so
4. Contingency.Pragmatic, indeed
5. Expansion.Alternative, instead
6. Expansion.Conjunction, also
7. Expansion.Instantiation, for example
8. Expansion.List, and
9. Expansion.Restatement, specifically
10. Temporal.Asynchronous, before
11. Temporal.Synchrony, when
ChatGPT: Expansion.Instantiation, for example

Query Template: Argument 1: Right away you notice the following things about a Philip Glass concert. Argument 2: It attracts people with funny hair. What is the relation path between Argument 1 and Argument 2? Select from the candidates.
1. Comparison -> Concession -> if
2. Comparison -> Contrast -> however
3. Contingency -> Cause -> so
4. Contingency -> Pragmatic -> indeed
5. Expansion -> Alternative -> instead
6. Expansion -> Conjunction -> also
7. Expansion -> Instantiation -> for example
8. Expansion -> List -> and
9. Expansion -> Restatement -> specifically
10. Temporal -> Asynchronous -> before
11. Temporal -> Synchrony -> when
ChatGPT: Expansion -> List -> and

Ground truth: Expansion.List

Table 20: Generation cases from three prompt templates for ChatGPT. The top one is the label-only template, the middle one is the label & connectives template, and the bottom is the structural path template.

Query Template: Argument 1: Grinned Griffith Peck, a trader in Shearson Lehman Hutton Inc.'s OTC department: "I tell you, this market acts healthy". Argument 2: Around him, scores of traders seemed to get a burst of energy. What is the relation label between Argument 1 and Argument 2? Select from the candidates.
1. Comparison.Concession
2. Comparison.Contrast
3. Contingency.Cause
4. Contingency.Pragmatic
5. Expansion.Alternative
6. Expansion.Conjunction
7. Expansion.Instantiation
8. Expansion.List
9. Expansion.Restatement
10. Temporal.Asynchronous
11. Temporal.Synchrony
ChatGPT: Contingency.Cause

Query Template: Argument 1: Grinned Griffith Peck, a trader in Shearson Lehman Hutton Inc.'s OTC department: "I tell you, this market acts healthy". Argument 2: Around him, scores of traders seemed to get a burst of energy. What is the relation and connective between Argument 1 and Argument 2? Select from the candidates.
1. Comparison.Concession, if
2. Comparison.Contrast, however
3. Contingency.Cause, so
4. Contingency.Pragmatic, indeed
5. Expansion.Alternative, instead
6. Expansion.Conjunction, also
7. Expansion.Instantiation, for example
8. Expansion.List, and
9. Expansion.Restatement, specifically
10. Temporal.Asynchronous, before
11. Temporal.Synchrony, when
ChatGPT: Contingency.Cause, so

Query Template: Argument 1: Grinned Griffith Peck, a trader in Shearson Lehman Hutton Inc.'s OTC department: "I tell you, this market acts healthy". Argument 2: Around him, scores of traders seemed to get a burst of energy. What is the relation path between Argument 1 and Argument 2? Select from the candidates.
1. Comparison -> Concession -> if
2. Comparison -> Contrast -> however
3. Contingency -> Cause -> so
4. Contingency -> Pragmatic -> indeed
5. Expansion -> Alternative -> instead
6. Expansion -> Conjunction -> also
7. Expansion -> Instantiation -> for example
8. Expansion -> List -> and
9. Expansion -> Restatement -> specifically
10. Temporal -> Asynchronous -> before
11. Temporal -> Synchrony -> when
ChatGPT: Temporal -> Synchrony -> when

Ground truth: Temporal.Synchrony

Table 21: Generation cases from three prompt templates for ChatGPT. The top one is the label-only template, the middle one is the label & connectives template, and the bottom is the structural path template.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
In the limitation section after the conclusion.

☑ A2. Did you discuss any potential risks of your work?
In the ethics statement section after the limitation section.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
In the abstract and introduction section

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

The used dataset details are in section 4.1 and appendix section A.1. The utilized software is cited in appendix section A.2.

☑ B1. Did you cite the creators of artifacts you used?
The used datasets are cited in section 4.1 and appendix section A.1. The utilized software is cited in appendix section A.2.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The license or terms for the used datasets are stated in Appendix A.1.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
The intended use for the used datasets is stated in Appendix A.1, we have not created any dataset from the existing dataset, and all dataset used is consistent with their intended use for research purposes.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
In the ethics statement section after the limitation section.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
The coverage of domains and the languages in the used datasets are stated in Appendix A.1.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
The relevant statistics in the used datasets are stated in Appendix A.1.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

The details of computational experiments is in section 4.2 and Appendix A.2.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
The details is in section 4.2, Appendix A.2 and A.6 .

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
The details of experimental setup is in section 4.2 and Appendix A.2 .

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
The details is in section 4.2 and Appendix A.2 .

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
The details of used existing packages is in Appendix A.2 .

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.