Annotation-Inspired Implicit Discourse Relation Classification with Auxiliary Discourse Connective Generation

Wei Liu and **Michael Strube**
Heidelberg Institute for Theoretical Studies gGmbH
{wei.liu, michael.strube}@h-its.org

Abstract

Implicit discourse relation classification is a challenging task due to the absence of discourse connectives. To overcome this issue, we design an end-to-end neural model to explicitly generate discourse connectives for the task, inspired by the annotation process of PDTB. Specifically, our model jointly learns to generate discourse connectives between arguments and predict discourse relations based on the arguments and the generated connectives. To prevent our relation classifier from being misled by poor connectives generated at the early stage of training while alleviating the discrepancy between training and inference, we adopt Scheduled Sampling to the joint learning. We evaluate our method on three benchmarks, PDTB 2.0, PDTB 3.0, and PCC. Results show that our joint model significantly outperforms various baselines on three datasets, demonstrating its superiority for the task.

1 Introduction

Discourse relations, such as *Cause* and *Contrast*, describe the logical relation between two text spans (Pitler et al., 2009). Recognizing discourse relations is beneficial for various NLP tasks, including coherence modeling (Lin et al., 2011), reading comprehension (Mihaylov and Frank, 2019), argumentation mining (Habernal and Gurevych, 2017; Hewett et al., 2019), and machine translation (Meyer, 2015; Longyue, 2019).

Discourse connectives (e.g., *but*, *as a result*) are words or phrases that signal the presence of a discourse relation (Pitler and Nenkova, 2009). They can be explicit, as in (1), or implicit, as in (2):

(1) [I refused to pay the cobbler the full $95]$_{Arg1}$ **because** [he did poor work.]$_{Arg2}$

(2) [They put the treasury secretary back on the board.]$_{Arg1}$ (**Implicit=However**) [There is doubt that the change would accomplish much.]$_{Arg2}$

When discourse connectives are explicitly present between arguments, classifying the sense of a discourse relation is straightforward. For example, Pitler and Nenkova (2009) proved that using only connectives in a text as features, the accuracy of 4-way explicit discourse relation classification on PDTB 2.0 can reach 85.8%. However, for implicit cases, there are no connectives to explicitly mark discourse relations, which makes implicit discourse relation classification challenging (Zhou et al., 2010; Shi et al., 2017). Existing work attempts to perform implicit discourse relation classification directly from arguments. They range from designing linguistically informed features from arguments (Lin et al., 2009; Pitler et al., 2009) to modeling interaction between arguments using neural networks (Lei et al., 2017; Guo et al., 2018). Despite their impressive performance, the absence of explicit discourse connectives makes the prediction extremely hard and hinders further improvement (Lin et al., 2014; Qin et al., 2017).

The huge performance gap between explicit and implicit classification (85.8% vs. 57.6%) (Liu and Li, 2016) motivates recent studies to utilize implicit connectives for the training process of implicit relation classifiers. For instance, Qin et al. (2017) developed an adversarial model to transfer knowledge from the model supplied with implicit connectives to the model without such information, while Kishimoto et al. (2020) proposed a multi-task learning framework to incorporate implicit connectives prediction as another training objective. However, we argue that these methods are suboptimal since connectives are still not explicitly present in input texts. This is demonstrated by Kishimoto et al. (2020), concluding that adding implicit connective prediction as a training objective provides only negligible gain for implicit relation classification on PDTB 2.0 (we empirically found the conclusion also held on the adversarial model).

In this paper, we design a novel end-to-end

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 15696–15712

model to leverage discourse connectives for the task of implicit discourse relation classification. The key inspiration is derived from the annotation process of implicit discourse relations in PDTB, which consists of inserting a connective that best conveys the inferred relation, and annotating the relation label based on both the inserted implicit connectives and contextual semantics (Prasad et al., 2008). We imitate this process by explicitly generating discourse connectives for the implicit relation classifier. Specifically, our model jointly learns to generate discourse connectives between arguments and predict discourse relations based on the arguments and the generated connectives. A potential drawback of this joint model is that the poorly generated connectives at the early stage of joint training may mislead the relation classifier. One possible solution is always feeding true connectives to the implicit relation classifier for training. But it leads to severe discrepancies between training and inference (Sporleder and Lascarides, 2008), since manually-annotated connectives are unavailable during evaluation (Prasad et al., 2008). To address this issue, we adopt Scheduled Sampling (Bengio et al., 2015) into our method. To be more specific, our relation classifier is first trained with hand-annotated implicit connectives and then gradually shifts to use generated connectives.

We evaluate our model[1] on two English corpora, PDTB 2.0 (Prasad et al., 2008), PDTB 3.0 (Webber et al., 2019), and a German corpus, PCC (Bourgonje and Stede, 2020), and compare it with other connective-enhanced approaches and existing state-of-the-art works. Results show that our method significantly outperforms those connective-enhanced baselines on three datasets while offering comparable performance to existing sota models.

In addition, we perform the first systematic analysis of different connective-enhanced models to investigate why our method works better. Our studies show that: (1) models learn to use connectives more effectively when putting connectives in the input rather than using them as training objectives; (2) end-to-end training can improve models' robustness to incorrectly-predicted connectives; (3) our method shows a better balance between arguments and connectives for relation prediction than other baselines. Finally, we show that connectives can effectively improve the predictive performance on frequent relations while failing on those with limited training instances.

2 Related Work

Implicit discourse relation classification, as a challenging part of shallow discourse parsing, has drawn much attention since the release of PDTB 2.0 (Prasad et al., 2008). Most of the work focused on predicting implicit relations directly from input arguments. For example, early statistical methods have put much effort into designing linguistically informed features from arguments (Pitler et al., 2009; Pitler and Nenkova, 2009; Lin et al., 2009; Rutherford and Xue, 2014). More recently, neural networks (Zhang et al., 2015; Kishimoto et al., 2018; Liu et al., 2020; Wu et al., 2022; Long and Webber, 2022) have been applied to learning useful semantic and syntactic information from arguments due to their strength in representation learning. Despite achieving impressive results, the absence of connectives makes their performance still lag far behind explicit discourse parsing.

The question of how to leverage discourse connectives for implicit discourse relation classification has received continued research attention. Zhou et al. (2010) proposed a pipeline method to investigate the benefits of connectives recovered from an n-gram language model for implicit relation recognition. Their results show that using recovered connectives as features can achieve comparable performance to a strong baseline. This pipeline-based method is further improved by following efforts, including integrating pre-trained models (Kurfalı and Östling, 2021; Jiang et al., 2021) and using prompt strategies (Xiang et al., 2022; Zhou et al., 2022). However, some works (Qin et al., 2017; Xiang and Wang, 2023) pointed out that pipeline methods suffer cascading errors. Recent studies have shifted to using end-to-end neural networks. Qin et al. (2017) proposed a feature imitation framework in which an implicit relation network is driven to learn from another neural network with access to connectives. Shi and Demberg (2019) designed an encoder-decoder model that generates implicit connectives from texts and learns a relation classifier using the representation of the encoder. Kishimoto et al. (2020) investigated a multi-task learning approach to predict connectives and discourse relations simultaneously. Our method is in line with those recent approaches exploiting connectives with an end-to-end neural network. The main difference is that those models

[1] https://github.com/liuwei1206/ConnRel

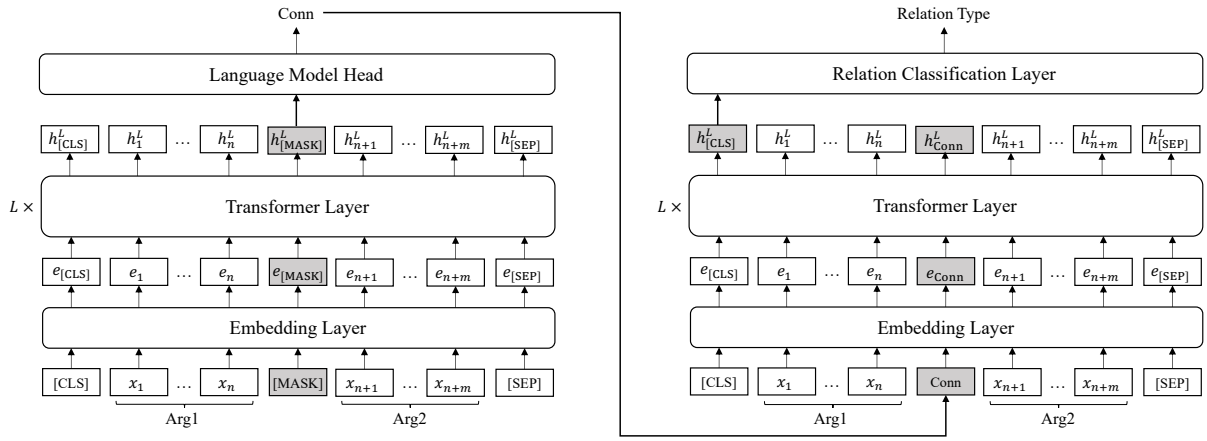

Figure 1: An overview of the proposed approach. The left part is the connective generation module which generates a connective at the masked position between arguments (Arg1, Arg2). The right part is the relation classification module which predicts the relation based on both arguments and the generated connective. We share the embedding layer and transformer blocks between two modules and train the whole model in an end-to-end manner.

focus on using implicit connectives in a non-input manner (i.e. they do not input implicit connectives as features but utilize them as another training signal), whereas our method explicitly generates connectives and inputs both arguments and the generated connectives into the relation classifier.

Our method can be viewed as a joint learning framework. Such a framework has been used to learn information exchange and reduce error propagation between related tasks (Zhang, 2018). Collobert et al. (2011) designed a unified neural model to perform tagging, chunking, and NER jointly. Søgaard and Goldberg (2016) refined this unified framework by putting low-level tasks supervised at lower layers. Miwa and Bansal (2016) presented an LSTM-based model to extract entities and the relations between them. Strubell et al. (2018) proposed a joint model for semantic role labeling (SRL), in which dependency parsing results were used to guide the attention module in the SRL task. Compared with these works, our joint learning framework is different in both motivation and design. For example, instead of simply sharing an encoder between tasks, we input the results of connective generation into the relation classifier.

3 Method

Inspired by the annotation process of PDTB, we explicitly generate discourse connectives for implicit relation classification. Following previous work (Lin et al., 2009), we use the gold standard arguments and focus on relation prediction. Figure 1 shows the overall architecture of our proposed model. It consists of two components: (1) generating a discourse connective between arguments; (2) predicting discourse relation based on arguments and the generated connective. In this section, we describe these two components in detail and show the challenges during training and our solutions.

Formally, let $X_1 = \{x_1, ..., x_n\}$ and $X_2 = \{x_{n+1}, ..., x_{n+m}\}$ be the two input arguments (Arg1 and Arg2) of implicit relation classification, where x_i denotes the i-th word in Arg1 and x_{n+j} denotes the j-th word in Arg2. We denote the relation between those two arguments as y. Similar to the setup in existing connective enhanced methods, each training sample (X_1, X_2, c, y) also includes an annotated implicit connective c that best expresses the relation. During the evaluation, only arguments (X_1, X_2) are available to the model.

3.1 Connective Generation

Connective generation aims to generate a discourse connective between two arguments (shown in the left part of Figure 1). We achieve this by using bidirectional masked language models (Devlin et al., 2019), such as RoBERTa. Specifically, we insert a [MASK] token between two arguments and generate a connective on the masked position.

Given a pair of arguments Arg1 and Arg2, we first concatenate a [CLS] token, argument Arg1, a [MASK] token, argument Arg2, and a [SEP] token into $\widetilde{X} = \{[\text{CLS}]\ X_1\ [\text{MASK}]\ X_2\ [\text{SEP}]\}$. For each token $\tilde{x}_i$ in $\widetilde{X}$, we convert it into the vector space by adding token, segment, and position embeddings, thus yielding input embeddings $E \in \mathbb{R}^{(n+m+3)\times d}$, where d is the hidden size. Then we input E into L stacked Transformer

blocks, and each Transformer layer acts as follows:

$$\begin{aligned} G &= \mathrm{LN}(H^{l-1} + \mathrm{MHAttn}(H^{l-1})) \\ H^{l} &= \mathrm{LN}(G + \mathrm{FFN}(G)) \end{aligned} \tag{1}$$

where H^l denotes the output of the l-th layer and $H^0 = E$; LN is layer normalization; MHAttn is the multi-head attention mechanism; FFN is a two-layer feed-forward network with ReLU as hidden activation function. To generate a connective on the masked position, we feed the hidden state of the [MASK] token after L Transformer layers into a language model head (LMHead):

$$\mathbf{p}^c = \mathrm{LMHead}(h^L_{[\mathrm{MASK}]}) \tag{2}$$

where $\mathbf{p}^c$ denotes the probabilities over the whole connective vocabulary. However, a normal LMHead can only generate one word without the capacity to generate multi-word connectives, such as "for instance". To overcome this shortcoming, we create several special tokens in LMHead's vocabulary to represent those multi-word connectives, and initialize their embedding with the average embedding of the contained single words. Taking "for instance" as an example, we create a token [for_instance] and set its embedding as $\mathrm{Average}(\mathrm{embed}(\text{"for"}), \mathrm{embed}(\text{"instance"}))$.

We choose cross-entropy as loss function for the connective generation module:

$$\mathcal{L}_{Conn} = -\sum_{i=0}^{N}\sum_{j=0}^{CN} C_{ij}\log(P^c_{ij}) \tag{3}$$

where C_i is the annotated implicit connective of the i-th sample with a one-hot scheme, CN is the total number of connectives.

3.2 Relation Classification

The goal of relation classification is to predict the implicit relation between arguments. Typically, it is solved using only arguments as input (Zhang et al., 2015; Kishimoto et al., 2018). In this work, we propose to predict implicit relations based on both input arguments and the generated connectives (shown in the right part of Figure 1).

First, we need to obtain a connective from the connective generation module. A straightforward way to do so is to apply the $\arg\max$ operation on the probabilities output by LMHead, i.e. $\mathrm{Conn} = \arg\max(\mathbf{p}^c)$. However, it is a non-differentiable process, which means the training signal of relation classification can not be propagated back to adjust the parameters of the connective generation module. Hence, we adopt the Gumbel-Softmax technique (Jang et al., 2017) for the task. The Gumbel-Softmax technique has been shown to be an effective approximation to the discrete variable (Shi et al., 2021). Therefore, we use

$$\begin{aligned} g &= -\log(-\log(\xi)),\ \xi \sim \mathrm{U}(0,1) \\ \mathbf{c}_i &= \frac{\exp((\log(p^c_i) + g_i)/\tau)}{\sum_j \exp((\log(p^c_j) + g_j)/\tau)} \end{aligned} \tag{4}$$

as the approximation of the one-hot vector of the generated connective on the masked position (denoted as Conn in Figure 1), where g is the Gumbel distribution, U is the uniform distribution, p^c_i is the probability of i-th connective output by the LMHead, $\tau \in (0, \infty)$ is a temperature parameter.

After we have obtained the generated connective "Conn", we concatenate it with arguments and construct a new input as $\bar{X} = \{[\mathrm{CLS}]\ X_1\ \mathrm{Conn}\ X_2\ [\mathrm{SEP}]\}$. This new form of input is precisely the same as the input in explicit discourse relation classification. We argue that the key to fully using connectives is to insert them into the input texts instead of treating them simply as a training objective. Like the connective generation module, we feed $\bar{X}$ into an Embedding layer and L stacked Transformer blocks. Note that we share the Embedding Layer and Transformers between connective generation and relation classification modules. Doing so can not only reduce the total memory for training the model but also prompt the interaction between two tasks. Finally, we feed the outputs of the L-th Transformer at [CLS] position to a relation classification layer:

$$\mathbf{p}^r = \mathrm{softmax}(\mathbf{W}_r h^L_{[\mathrm{CLS}]} + \mathbf{b}_r) \tag{5}$$

where $\mathbf{W}_r$ and $\mathbf{b}_r$ are learnable parameters. Similarly, we use cross-entropy for training, and the loss is formulated as:

$$\mathcal{L}_{Rel} = -\sum_{i=0}^{N}\sum_{j=0}^{RN} Y_{ij}\log(P^r_{ij}) \tag{6}$$

where Y_i is the ground truth relation of the i-th sample with a one-hot scheme, RN is the total number of relations.

3.3 Training and Evaluation

To jointly train those two modules, we use a multi-task loss:

$$\mathcal{L} = \mathcal{L}_{Conn} + \mathcal{L}_{Rel} \tag{7}$$

A potential issue of this training is that poorly generated connectives at an early stage of joint training

Algorithm 1 Scheduled Sampling in Training

Input: relation classifier $\mathbf{RelCls}$, arguments $\mathrm{X}_1, \mathrm{X}_2$, annotated connective $\mathrm{true\_conn}$, generated connective $\mathrm{gene\_conn}$, training step t, hyperparameter in decay k

Output: logits

$\mathrm{p} = \mathrm{random}()$ ▷ $[0.0, 1.0)$
$\epsilon_\mathrm{t} = \frac{\mathrm{k}}{\mathrm{k}+\exp(\mathrm{t}/\mathrm{k})}$
if $\mathrm{p} < \epsilon_\mathrm{t}$ **then**
    $\mathrm{logits} = \mathbf{RelCls}(\mathrm{X}_1, \mathrm{X}_2, \mathrm{true\_conn})$
else
    $\mathrm{logits} = \mathbf{RelCls}(\mathrm{X}_1, \mathrm{X}_2, \mathrm{gene\_conn})$
end if

may mislead the relation classifier. One possible solution is always providing manually annotated implicit connectives to the relation classifier, similar to Teacher Forcing (Ranzato et al., 2016). But this might lead to a severe discrepancy between training and inference since manually annotated connectives are not available during inference. We solve those issues by introducing Scheduled Sampling (Bengio et al., 2015) into our method. Scheduled Sampling is designed to sample tokens between gold references and model predictions with a scheduled probability in seq2seq models. We adopt it into our training by sampling between manually-annotated and the generated connectives. Specifically, we use the inverse sigmoid decay (Bengio et al., 2015), in which probability of sampling manually annotated connectives at the t-th training step is calculated as follows:

$$\epsilon_t = \frac{k}{k + \exp(t/k)} \tag{8}$$

where $k \geq 1$ is a hyper-parameter to control the convergence speed. In the beginning, training is similar to Teacher Forcing due to $\epsilon_t \approx 1$. As the training step t increases, the relation classifier gradually uses more generated connectives, and eventually uses only generated ones (identical to the evaluation setting) when $\epsilon_t \approx 0$. We show the sampling process during training in Algorithm 1.

During inference, we generate a connective Conn through $\arg\max(\mathbf{p^c})$, feed the generated Conn and arguments into the relation classifier, and choose the relation type that possesses the maximum value in $\mathbf{p}^r$.

4 Experiments

We carry out a set of experiments to investigate the effectiveness of our method across different corpora and dataset splittings. In addition, we perform analyses showing that our model learns a better balance between using connectives and arguments than baselines.

4.1 Experimental Settings

Datasets. We evaluate our model on two English corpora, PDTB 2.0 (Prasad et al., 2008), PDTB 3.0 (Webber et al., 2019), and a German corpus, PCC (Bourgonje and Stede, 2020). In PDTB, instances are annotated with senses from a three-level sense hierarchy. We follow previous works (Ji and Eisenstein, 2015; Kim et al., 2020) to use top-level 4-way and second-level 11-way classification for PDTB 2.0, and top-level 4-way and second-level 14-way for PDTB 3.0. As for the dataset split, we adopt two different settings for both PDTB 2.0 and PDTB 3.0. The first one is proposed by Ji and Eisenstein (2015), where sections 2-20, sections 0-1, and sections 21-22 are used as training, development, and test set. The second one is called section-level cross-validation (Kim et al., 2020), in which 25 sections are divided into 12 folds with 2 validation, 2 test, and 21 training sections. There are over one hundred connectives in PDTB (e.g., 102 in PDTB 2.0), but some rarely occur (e.g., only 7 for "next" in PDTB 2.0). To reduce the complexity of connective generation and ensure each connective has sufficient training data, we only consider connectives with a frequency of at least 100 in the experiments. PCC is a German corpus following the annotation guidelines of PDTB. For this corpus, we only use the second-level 8-way classification since the distribution of top-level relations is highly uneven (Bourgonje, 2021). A more detailed description and statistics of the datasets are given in Appendix A.

Implementation Details. We implement our model using the Pytorch library. The bidirectional masked language model used in our work is $\mathrm{RoBERTa}_{\mathrm{base}}$, which is initialized with the pre-trained checkpoint from Huggingface. For hyper-parameter configurations, we mainly follow the settings in RoBERTa (Liu et al., 2019). We use the AdamW optimizer with an initial learning rate of 1e-5, a batch size of 16, and a maximum epoch number of 10 for training. Considering the training variability in PDTB, we report the mean performance of 5 random restarts for the "Ji" splits and that of the section-level cross-validation (Xval) like Kim et al. (2020). For PCC, we conduct a 5-fold

	Level1 4-way				Level2 11-way			
	Ji		Xval		Ji		Xval	
Models	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Liu et al. (2020)	$69.06_{0.43}$	$63.39_{0.56}$	-	-	$58.13_{0.67}$	-	-	-
Kim et al. (2020)	66.30	56.00	-	-	$54.73_{0.79}$	-	$52.98_{0.29}$	-
Wu et al. (2022)	71.18	63.73	-	-	60.33	40.49	-	-
Zhou et al. (2022)	70.84	64.95	-	-	60.54	41.55	-	-
Long and Webber (2022)	72.18	**69.60**	-	-	61.69	**49.66**	-	-
RoBERTa	$68.61_{0.73}$	$60.89_{0.19}$	$68.66_{1.29}$	$60.49_{1.86}$	$58.84_{0.48}$	$39.31_{0.83}$	$55.40_{1.65}$	$36.51_{2.75}$
RoBERTaConn	$55.34_{0.39}$	$37.47_{2.27}$	$54.28_{2.12}$	$34.71_{2.75}$	$31.97_{2.75}$	$17.10_{2.81}$	$32.12_{2.63}$	$17.91_{2.12}$
Adversarial	$69.43_{0.70}$	$62.44_{0.61}$	$69.13_{1.14}$	$60.63_{1.47}$	$57.63_{1.10}$	$38.81_{2.25}$	$54.43_{1.79}$	$36.79_{2.24}$
Multi-Task	$70.82_{0.72}$	$63.79_{0.82}$	$70.02_{1.40}$	$62.19_{1.84}$	$60.21_{0.94}$	$39.75_{0.70}$	$56.85_{1.13}$	$36.83_{2.42}$
Pipeline	$71.01_{0.89}$	$64.65_{1.03}$	$69.12_{1.03}$	$61.65_{0.89}$	$59.42_{0.54}$	$40.84_{0.39}$	$55.24_{1.72}$	$37.03_{2.83}$
Our Model	$\mathbf{74.59}_{0.44}$	$68.64_{0.67}$	$\mathbf{71.33}_{1.25}$	$\mathbf{63.84}_{1.96}$	$\mathbf{62.75}_{0.59}$	$42.36_{0.38}$	$\mathbf{57.98}_{1.22}$	$\mathbf{39.05}_{3.53}$

Table 1: Results on PDTB **2.0**. Subscripts are the standard deviation of the mean performance.

cross-validation (Xval) on this corpus due to its limited number of data. We use standard accuracy (Acc, %) and F1-macro (F1, %) as evaluation metrics. We show more detailed settings and hyperparameters in Appendix B.

Baselines. To demonstrate the effectiveness of our model, we compare it with state-of-the-art connective-enhanced methods and several variants of our model:

- **RoBERTa**. Finetune RoBERTa for implicit relation classification. Only arguments (Arg1, Arg2) are input for training without using any implicit discourse connective information.
- **RoBERTaConn**. A variant of the RoBERTa baseline. During training, we feed both arguments and annotated connectives, i.e., (Arg1, Arg2, true_conn), to RoBERTa. During inference, only arguments (Arg1, Arg2) are input to the model.
- **Adversarial**. An adversarial-based connective enhanced method (Qin et al., 2017), in which an implicit relation network is driven to learn from another neural network with access to connectives. We replace its encoder with $\mathrm{RoBERTa}_{\mathrm{base}}$ for a fair comparison.
- **Multi-Task**. A multi-task framework for implicit relation classification (Kishimoto et al., 2020), in which connective prediction is introduced as another training task. We equip it with the same $\mathrm{RoBERTa}_{\mathrm{base}}$ as our method.
- **Pipeline**. A pipeline variant of our method, in which we first train a connective generation model, then learn a relation classifier with arguments and the generated connectives. Note that these two modules are trained separately.

Further, we compare our method against previous state-of-the-art models on each corpus.

4.2 Overall Results

PDTB 2.0. Table 1 shows the experimental results on PDTB 2.0. RoBERTaConn shows a much worse performance than the RoBERTa baseline on this corpus, indicating that simply feeding annotated connectives to the model causes a severe discrepancy between training and evaluation. This is also somewhat in accord with Sporleder and Lascarides (2008), which shows that models trained on explicitly-marked examples generalize poorly to implicit relation identification. Discourse connective-enhanced models, including Adversarial, Multi-Task, Pipeline and Our Method, achieve better performance than the RoBERTa baseline. This demonstrates that utilizing the annotated connectives information for training is beneficial for implicit relation classification. The improvement of Adversarial and Multi-task over the RoBERTa baseline is limited and unstable. We argue this is because they do not exploit connectives in the way of input features but treat them as training objectives, thus limiting connectives' contributions to implicit relation classification. Pipeline also shows limited performance gain over the baseline. We speculate that this is due to its pipeline setting (i.e. connective generation $\rightarrow$ relation classification), which propagates errors in connective generation to relation classification (Qin et al., 2017). Compared to the above connective-enhanced models, our method's improvement over the RoBERTa baseline is bigger, which suggests that our approach is more efficient in utilizing connectives. To further show the efficiency of our method, we compare it against previous state-of-

	Level1 4-way (**PDTB 3.0**)				Level2 14-way (**PDTB 3.0**)				Level2 8-way (**PCC**)	
	Ji		Xval		Ji		Xval		Xval	
Models	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Kim et al. (2020)	71.30	64.80	-	-	-	-	$60.78_{0.24}$	-	-	-
Xiang et al. (2022)	74.36	69.91	-	-	-	-	-	-	-	-
Long and Webber (2022)	75.31	70.05	-	-	64.68	**57.62**	-	-	-	-
RoBERTa	$73.51_{0.69}$	$67.98_{0.97}$	$73.42_{0.90}$	$67.54_{1.40}$	$63.32_{0.40}$	$52.49_{1.26}$	$62.65_{1.32}$	$53.19_{1.20}$	$35.80_{1.13}$	$15.08_{0.97}$
RoBERTaConn	$51.74_{0.76}$	$41.45_{0.69}$	$53.90_{1.71}$	$39.39_{2.74}$	$33.67_{1.78}$	$25.40_{2.11}$	$36.68_{2.39}$	$28.18_{4.11}$	$30.30_{2.86}$	$12.62_{2.06}$
Adversarial	$73.83_{0.28}$	$68.60_{0.75}$	$73.30_{1.32}$	$67.23_{1.85}$	$63.00_{0.48}$	$54.28_{1.76}$	$62.12_{1.46}$	$53.85_{1.46}$	$35.02_{3.18}$	$18.48_{1.51}$
Multi-Task	$74.97_{0.70}$	$69.67_{0.76}$	$73.83_{0.94}$	$68.04_{1.30}$	$64.52_{0.31}$	$53.12_{0.63}$	$62.81_{1.36}$	$53.07_{1.40}$	$40.48_{1.47}$	$21.22_{2.01}$
Pipeline	$74.54_{0.22}$	$69.19_{0.60}$	$73.70_{0.89}$	$68.31_{1.78}$	$63.98_{0.63}$	$52.95_{0.48}$	$63.07_{1.70}$	$53.43_{1.63}$	$42.97_{3.48}$	$22.66_{1.20}$
Our Model	$\mathbf{76.23}_{0.19}$	$\mathbf{71.15}_{0.47}$	$\mathbf{75.41}_{0.89}$	$\mathbf{70.06}_{1.72}$	$\mathbf{65.51}_{0.41}$	$54.92_{0.81}$	$\mathbf{64.59}_{1.21}$	$\mathbf{55.26}_{1.32}$	$\mathbf{44.54}_{3.06}$	$\mathbf{26.93}_{2.06}$

Table 2: Results on PDTB **3.0** and **PCC**. Subscripts are the standard deviation of the mean performance.

the-art models on PDTB 2.0 (Liu et al., 2020; Kim et al., 2020; Wu et al., 2022; Zhou et al., 2022; Long and Webber, 2022). The first block of Table 1 shows the results of those models, from which we observe that our model outperforms most of them, especially on accuracy, achieving the best results on this corpus. The only exception is that the F1-score of our method lags behind Long and Webber (2022), particularly on level2 classification. This is because our method cannot predict several fine-grained relations (see Section 4.4), such as Comparison.Concession, which leads to the low averaged F1 at the label-level.

PDTB 3.0 / PCC. Results on PDTB 3.0 and PCC are shown in Table 2. Similar to the results on the PDTB 2.0 corpus, simply feeding connectives for training (RoBERTaConn) hurts the performance, especially on the Level2 classification of PDTB 3.0. Adversarial and Multi-Task perform better than the RoBERTa baseline, although their improvement is limited. Despite suffering cascading errors, Pipeline shows comparative and even better results than Adversarial and Multi-Task on the two corpora. This indicates the advantage of utilizing connectives as input features rather than a training objective, particularly on PCC. Consistent with the results on PDTB 2.0, our method outperforms Adversarial, Multi-task, and Pipeline on both datasets, demonstrating the superiority of inputting connectives to the relation classifier in an end-to-end manner and also showing that it works well on different languages. We further compare our method with three existing sota models on PDTB 3.0, Kim et al. (2020), Xiang et al. (2022), and Long and Webber (2022). Results in Table 2 show that our approach performs better than these three models.

4.3 Performance Analysis

To figure out why our model works well, we first perform analyses on its behavior answering two questions: (1) whether it really benefits from discourse connectives; (2) whether it can also make correct predictions when connectives are missing. We then investigate the relation classifier's performance in the different models when connectives are correctly and incorrectly generated (or predicted).

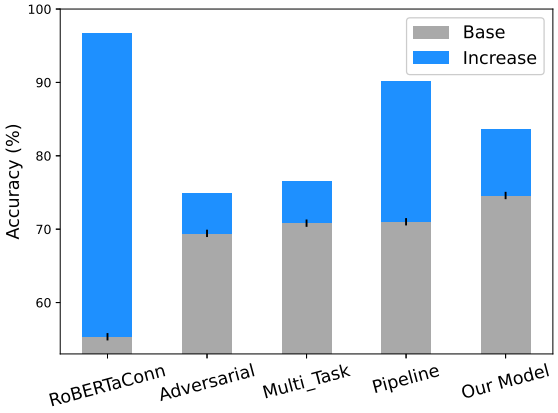

Figure 2: Level1 classification results on PDTB 2.0 (Ji split) when annotated connectives are fed to connective-enhanced models. "Increase" denotes performance gain compared to the model with default settings ("Base").

We perform the first analysis by replacing the generated connectives in our model with manually-annotated ones[2], and compare its performance before and after this setup. Intuitively, if our model benefits from discourse connectives, accuracy and F1-macro should increase after the change. For comparison, we apply the same setup to other connective-enhanced models. We conduct experiments[3] on the Level1 classification of PDTB 2.0 (Ji split), and show the accuracy results in Figure 2. As expected, our model's performance shows a substantial improvement, demonstrating that it does learn to use discourse connectives for implicit relation classification. Other connective-enhanced models also perform better in such a setup but with

[2]In PDTB 2.0 and PDTB 3.0, each instance contains annotated implicit connectives, making this analysis possible.

[3]We show more detailed results and also case studies in Appendix C.

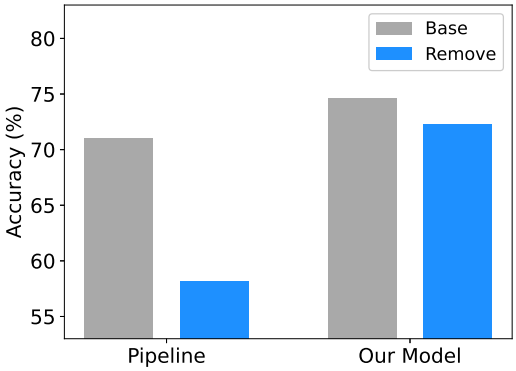

Figure 3: Level1 classification results on PDTB 2.0 (Ji split). "Remove" denotes the generated connectives are removed from the original model ("Base").

a different degree of gain. Specifically, models that use connectives as input features during training (RoBERTaConn, Pipeline, and Our Method) show more increase and have higher upper bounds than models that use connectives as training objectives (Adversarial and Multi-Task). This aligns with our assumption that putting connectives in the input is more efficient for a model learning to use discourse connectives for implicit relation classification than treating them as training objectives. However, inputting connectives for training can lead to another severe issue, i.e., the model relies too much on connectives for prediction. For instance, the RoBERTaConn's performance will drop from 96.69% to 55.34% when manually-annotated connectives are not available.

To probe whether our model suffers such an issue, we perform the second analysis by removing the generated connectives in our model and observing changes in its performance. The same setting is applied to Pipeline for comparison. Figure 3 shows the Level1 classification results[3] on PDTB 2.0 (Ji split). Both models see a performance drop but still outperform RoBERTaConn. This is because these two models' relation classifiers input the generated connectives rather than the annotated ones for training, alleviating their reliance on connectives. The decrease of Our Method (74.59% $\rightarrow$ 72.27%) is much smaller than that of Pipeline (71.01% $\rightarrow$ 58.15%). We speculate that the end-to-end training enables our model to learn a good balance between arguments and discourse connectives for relation classification. By contrast, Pipeline fails to do so due to the separate training of connectives generation and relation classification.

Finally, we show in Table 3 the results of relation classifiers in Multi-Task, Pipeline, and Our method[4] on PDTB 2.0 when connectives are correctly and incorrectly generated or predicted. Note that these three models' results are not directly comparable in the correct and incorrect groups since their predictions on connectives are different[3] (not overlap). To solve this, we calculate the performance gain of each model over the RoBERTa baseline and compare them from the gain perspective. When connectives are correctly generated, Pipeline and Our Model outperform the RoBERTa baseline by more than 10% in accuracy, while Multi-task's improvement is only 6.9%. This suggests that Pipeline and Our Model utilize connectives more efficiently than Multi-Task. On the other hand, when the connectives' prediction is incorrect, Pipeline's performance is worse than the RoBERTa baseline by 1.64%. Compared to it, Multi-task and Our Method achieve comparable performance to RoBERTa, showing good robustness when exposed to incorrect connectives. Despite achieving better results than baselines in both groups, our model performs significantly worse in the incorrect connective group than in the correct one. This indicates that its major performance bottleneck originates from the incorrectly generated connectives. A possible improvement is first pre-training our model on a large explicit connectives corpus, like Sileo et al. (2019). By doing so, the connective generation module may generate more correct connectives, thus improving classification performance, which we leave for future work.

Models	Correct Group	Incorrect Group
$\text{Base}_{\text{Multi-Task}}$	83.67	59.82
Multi-Task	90.60(+6.93)	59.88(+0.06)
$\text{Base}_{\text{Pipeline}}$	78.87	61.46
Pipeline	89.29(+10.4)	59.81(-1.64)
$\text{Base}_{\text{Our Model}}$	80.28	60.56
Our Model	94.04(+13.8)	62.22(+1.66)

Table 3: Level1 classification results on PDTB 2.0 (Ji split) when connectives are correctly and incorrectly generated (or predicted). "+" and "-" denote the increase and decrease compared to the RoBERTa baseline (Base).

4.4 Relation Analysis

We investigate which relations benefit from the joint training of connective generation and relation classification and compare it with other baselines. Table 4 shows different models' F1-score for each second-level sense of PDTB 2.0 (Ji split). Generally, connectives benefit the prediction of most

[4]This analysis is not performed on other models (e.g., Adversarial) because they don't generate or predict connectives.

Labels	RoBERTa	Adversarial	Multi-Task	Pipeline	Our Model
Temporal.Asynchronous	54.62	55.01	58.37	55.69	**59.48**
Temporal.Synchrony	00.00	**06.03**	00.00	04.00	00.00
Contingency.Cause	60.03	59.00	64.24	65.40	**66.35**
Contingency.Pragmatic cause	00.00	**05.00**	00.00	00.00	00.00
Comparison.Contrast	60.44	58.20	61.73	60.78	**65.75**
Comparison.Concession	00.00	01.14	00.00	**01.82**	00.00
Expansion.Conjunction	56.03	53.26	**58.94**	54.79	57.04
Expansion.Instantiation	74.07	72.85	**74.12**	70.76	73.87
Expansion.Restatement	57.87	56.94	59.68	57.75	**60.94**
Expansion.Alternative	49.06	44.76	**54.82**	43.96	51.13
Expansion.List	18.07	11.68	11.43	**29.96**	25.47

Table 4: F1 results for each second-level relation of PDTB 2.0.

	PDTB 2.0		PDTB 3.0	
Models	Acc	F1	Acc	F1
Our Model	**74.59**	**68.64**	**76.23**	**71.15**
- SS	73.42	66.68	75.87	70.68
- SS, $\mathcal{L}_{Conn}$	70.63	63.43	74.58	69.17
RoBERTa	68.61	60.89	73.51	67.98

Table 5: Ablation study for **S**cheduled **S**ampling and connective generation loss $\mathcal{L}_{Conn}$.

relation types, especially in Multi-Task, Pipeline, and Our Method. For example, these three models outperform the RoBERTa baseline by more than 4% in the F1-score on the Contingency.Cause relation. On some relations, such as Expansion.Instantiation, connective-enhanced models show different tendencies, with some experiencing improvement while others drop. Surprisingly, all models fail to predict Temporal.Synchrony, Contingency.Pragmatic cause, and Comparison.Concession despite using manually-annotated connectives during training. We speculate this is caused by their limited number of training instances, making models tend to predict other frequent labels. One feasible solution to this issue is Contrastive Learning (Chen et al., 2020), which has been shown to improve the predictive performance of these three relations (Long and Webber, 2022). We leave integrating Contrastive Learning with our method to future work.

4.5 Ablation Study

We conduct ablation studies to evaluate the effectiveness of Scheduled Sampling (SS) and the Connective generation loss $\mathcal{L}_{Conn}$. To this end, we test the performance of our method by first removing SS and then removing $\mathcal{L}_{Conn}$. Note that removing $\mathcal{L}_{Conn}$ means that our whole model is trained with only gradients from $\mathcal{L}_{Rel}$.

Table 5 shows the Level1 classification results on PDTB 2.0 and PDTB 3.0 (Ji split). We can observe from the table that eliminating any of them would hurt the performance, showing their essential to achieve good performance. Surprisingly, our model training with only $\mathcal{L}_{Rel}$ performs much better than the RoBERTa baseline. This indicates that the performance gain of our full model comes not only from the training signals provided by manually-annotated connectives but also from its well-designed structure inspired by PDTB's annotation (i.e. the connective generation module and relation prediction module).

5 Conclusion

In this paper, we propose a novel connective-enhanced method for implicit relation classification, inspired by the annotation of PDTB. We introduce several key techniques to efficiently train our model in an end-to-end manner. Experiments on three benchmarks demonstrate that our method consistently outperforms various baseline models. Analyses of the models' behavior show that our approach can learn a good balance between using arguments and connectives for implicit discourse relation prediction.

6 Limitations

Despite achieving good performance, there are some limitations in our study. The first is how to handle ambiguous instances in the corpus. 3.45% of the implicit data in PDTB 2.0 and 5% in PDTB 3.0 contains more than one label. Currently, we follow previous work and simply use the first label for training. But there might be a better solution to handle those cases. Another is the required time for training. To mimic the annotation process of PDTB, our model needs to pass through the embedding layer and transformers twice, so it takes more

time to train than the RoBERTa baseline. However, our training time is shorter than Pipeline and Adversarial due to those two models' pipeline setup and adversarial training strategy. Also, note that our method has a similar number of parameters to the RoBERTa baseline since we share embedding layers and transformers between the connection generation and relation classification modules in our approach. Therefore, the memory required to train our model is not much different from that required to train the RoBERTa baseline.

Acknowledgements

The authors would like to thank the three anonymous reviewers for their comments. We also thank Xiyan Fu for her valuable feedback on earlier drafts of this paper. This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany. The first author has been supported by a Heidelberg Institute for Theoretical Studies Ph.D. scholarship.

References

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28.

Peter Bourgonje. 2021. *Shallow Discourse Parsing for German*. Doctoral Thesis, Universität Potsdam.

Peter Bourgonje and Manfred Stede. 2020. The Potsdam commentary corpus 2.2: Extending annotations for shallow discourse parsing. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1061–1066, Marseille, France. European Language Resources Association.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fengyu Guo, Ruifang He, Di Jin, Jianwu Dang, Longbiao Wang, and Xiangang Li. 2018. Implicit discourse relation recognition using neural tensor network with interactive attention and sparse learning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 547–558, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ivan Habernal and Iryna Gurevych. 2017. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 43(1):125–179.

Freya Hewett, Roshan Prakash Rane, Nina Harlacher, and Manfred Stede. 2019. The utility of discourse parsing features for predicting argumentation structure. In *Proceedings of the 6th Workshop on Argument Mining*, pages 98–103, Florence, Italy. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017*.

Yangfeng Ji and Jacob Eisenstein. 2015. One vector is not enough: Entity-augmented distributed semantics for discourse relations. *Transactions of the Association for Computational Linguistics*, 3:329–344.

Congcong Jiang, Tieyun Qian, Zhuang Chen, Kejian Tang, Shaohui Zhan, and Tao Zhan. 2021. Generating pseudo connectives with mlms for implicit discourse relation recognition. In *PRICAI 2021: Trends in Artificial Intelligence*, pages 113–126, Cham. Springer International Publishing.

Najoung Kim, Song Feng, Chulaka Gunasekara, and Luis Lastras. 2020. Implicit discourse relation classification: We need to talk about evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5404–5414, Online. Association for Computational Linguistics.

Yudai Kishimoto, Yugo Murawaki, and Sadao Kurohashi. 2018. A knowledge-augmented neural network model for implicit discourse relation classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 584–595, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yudai Kishimoto, Yugo Murawaki, and Sadao Kurohashi. 2020. Adapting BERT to implicit discourse relation classification with a focus on discourse connectives. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1152–1158, Marseille, France. European Language Resources Association.

Murathan Kurfalı and Robert Östling. 2021. Let's be explicit about that: Distant supervision for implicit discourse relation classification via connective prediction. In *Proceedings of the 1st Workshop on Understanding Implicit and Underspecified Language*,

pages 1–10, Online. Association for Computational Linguistics.

Wenqiang Lei, Xuancong Wang, Meichun Liu, Ilija Ilievski, Xiangnan He, and Min-Yen Kan. 2017. Swim: A simple word interaction model for implicit discourse relation recognition. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4026–4032.

Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing implicit discourse relations in the Penn Discourse Treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 343–351, Singapore. Association for Computational Linguistics.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2011. Automatically evaluating text coherence using discourse relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 997–1006, Portland, Oregon, USA. Association for Computational Linguistics.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, 20(2):151–184.

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 *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3830–3836. International Joint Conferences on Artificial Intelligence Organization. Main track.

Yang Liu and Sujian Li. 2016. Recognizing implicit discourse relations via repeated reading: Neural networks with multi-level attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1233, Austin, Texas. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Wanqiu Long and Bonnie Webber. 2022. Facilitating contrastive learning of discourse relational senses by exploiting the hierarchy of sense relations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10704–10716, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wang Longyue. 2019. *Discourse-aware neural machine translation*. Ph.D. thesis, Dublin City University.

Thomas Meyer. 2015. *Discourse-level features for statistical machine translation*. Ph.D. thesis, École polytechnique fédérale de Lausanne (EPFL).

Todor Mihaylov and Anette Frank. 2019. Discourse-aware semantic self-attention for narrative reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2541–2552, Hong Kong, China. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Emily Pitler, Annie Louis, and Ani Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 683–691, Suntec, Singapore. Association for Computational Linguistics.

Emily Pitler and Ani Nenkova. 2009. Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 13–16, Suntec, Singapore. Association for Computational Linguistics.

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

Lianhui Qin, Zhisong Zhang, Hai Zhao, Zhiting Hu, and Eric Xing. 2017. Adversarial connective-exploiting networks for implicit discourse relation classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1006–1017, Vancouver, Canada. Association for Computational Linguistics.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016*.

Attapol Rutherford and Nianwen Xue. 2014. Discovering implicit discourse relations through brown cluster pair representation and coreference patterns. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 645–654, Gothenburg, Sweden. Association for Computational Linguistics.

Jihao Shi, Xiao Ding, Li Du, Ting Liu, and Bing Qin. 2021. Neural natural logic inference for interpretable question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3673–3684, Online and

Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wei Shi and Vera Demberg. 2019. Learning to explicitate connectives with Seq2Seq network for implicit discourse relation classification. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 188–199, Gothenburg, Sweden. Association for Computational Linguistics.

Wei Shi, Frances Yung, Raphael Rubino, and Vera Demberg. 2017. Using explicit discourse connectives in translation for implicit discourse relation classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 484–495, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Damien Sileo, Tim Van De Cruys, Camille Pradel, and Philippe Muller. 2019. Mining discourse markers for unsupervised sentence representation learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3477–3486, Minneapolis, Minnesota. Association for Computational Linguistics.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235, Berlin, Germany. Association for Computational Linguistics.

Caroline Sporleder and Alex Lascarides. 2008. Using automatically labelled examples to classify rhetorical relations: an assessment. *Natural Language Engineering*, 14(3):369–416.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Bonnie Webber, Rashmi Prasad, Alan Lee, and Aravind Joshi. 2019. The Penn Discourse TreeBank 3.0 annotation manual. *Philadelphia, University of Pennsylvania*, 35:108.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11486–11494.

Wei Xiang and Bang Wang. 2023. A survey of implicit discourse relation recognition. *ACM Computing Surveys*, 55(12):1–34.

Wei Xiang, Zhenglin Wang, Lu Dai, and Bang Wang. 2022. ConnPrompt: Connective-cloze prompt learning for implicit discourse relation recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 902–911, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Attapol Rutherford, Bonnie Webber, Chuan Wang, and Hongmin Wang. 2016. CoNLL 2016 shared task on multilingual shallow discourse parsing. In *Proceedings of the CoNLL-16 shared task*, pages 1–19, Berlin, Germany. Association for Computational Linguistics.

Biao Zhang, Jinsong Su, Deyi Xiong, Yaojie Lu, Hong Duan, and Junfeng Yao. 2015. Shallow convolutional neural network for implicit discourse relation recognition. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2230–2235, Lisbon, Portugal. Association for Computational Linguistics.

Yue Zhang. 2018. Joint models for NLP. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, Melbourne, Australia. Association for Computational Linguistics.

Hao Zhou, Man Lan, Yuanbin Wu, Yuefeng Chen, and Meirong Ma. 2022. Prompt-based connective prediction method for fine-grained implicit discourse relation recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3848–3858, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhi Min Zhou, Man Lan, Zheng Yu Niu, Yu Xu, and Jian Su. 2010. The effects of discourse connectives prediction on implicit discourse relation recognition. In *Proceedings of the SIGDIAL 2010 Conference*, pages 139–146, Tokyo, Japan. Association for Computational Linguistics.

	PDTB 2.0	PDTB 3.0
L1	Comparison	Comparison
	Contingency	Contingency
	Expansion	Expansion
	Temporal	Temporal
L2	Comparison.Concession	Comparison.Concession
	Comparison.Contrast	Comparison.Contrast
	Contingency.Cause	Contingency.Cause
	Contingency.Pragmatic cause	Contingency.Cause+Belief
	Expansion.Conjunction	Contingency.Condition
	Expansion.Instantiation	Contingency.Purpose
	Expansion.Alternative	Expansion.Conjunction
	Expansion.List	Expansion.Equivalence
	Expansion.Restatement	Expansion.Instantiation
	Temporal.Asynchronous	Expansion.Level-of-detail
	Temporal.Synchrony	Expansion.Manner
		Expansion.Substitution
		Temporal.Asynchronous
		Temporal.Synchronous

Table 6: Top-level (L1) and second-level (L2) relations of PDTB 2.0 and PDTB 3.0 used in our experiments.

	Train	Dev	Test
PDTB 2.0	12632	1183	1046
PDTB 3.0	17085	1653	1474

Table 7: Dataset statistics for the "Ji" split.

A Data Description

The Penn Discourse TreeBank (PDTB) is the most common corpus for the task of implicit discourse relation classification. The annotation of this corpus follows a specific strategy, which consists of inserting a connective that best conveys the inferred relation, and annotating the relation label based on both the inserted implicit connectives and contextual semantics. Prasad et al. (2008) claimed that this annotation strategy significantly improves the inter-annotator agreement. PDTB has two widely used versions, PDTB 2.0 (Prasad et al., 2008) and PDTB 3.0 (Webber et al., 2019). In both versions, instances are annotated with senses[5] from a three-level sense hierarchy. We follow previous work (Ji and Eisenstein, 2015; Kim et al., 2020) to use top-level 4-way and second-level 11-way classification for PDTB 2.0, and top-level 4-way and second-level 14-way for PDTB 3.0, and show these relations in Table 6. We show the statistics information of Ji and Eisenstein (2015) and Kim et al. (2020) in Tables 7 and 8, respectively.

The Potsdam Commentary Corpus (PCC) is a German corpus constructed following the annotation guideline of PDTB (Bourgonje and Stede, 2020). In this dataset, relations are also organized in a three-level hierarchy structure. However, this corpus is relatively small, containing only 905 implicit data, and the distribution of its relations is highly uneven, especially the top-level relations. For example, the "Expansion" (540) and "Contingency" (246) account for more than 86% of the data among all top-level relations. Bourgonje (2021) concluded that two of four relations were never predicted in his classifier due to the highly uneven distribution of the top-level relation data. Therefore, we only use the second-level relations in our experiments. Furthermore, we use a similar setup to PDTBs for PCC, considering only relations whose frequency is not too low (over 10 in our setting). The final PCC used in our experiments contains 891 data covering 8 relations (shown in Table 9). As for connectives, here, we only consider connectives with a frequency of at least 5 due to the limited size of this corpus.

[5]Some instances in PDTB have more than one label. We follow previous work to use the first label for training. While evaluating, a prediction is regarded as correct if it matches one of the annotated labels (Xue et al., 2016).

fold	splitting	PDTB 2.0	PDTB 3.0
1	0-1 / 2-22 / 23-24	1183 / 13678 / 1192	1653 / 18559 / 1615
2	2-3 / 4-24 / 0-1	1154 / 13716 / 1183	1579 / 18595 / 1653
3	4-5 / 6-1 / 2-3	1527 / 13372 / 1154	2039 / 18209 / 1579
4	6-7 / 8-3 / 4-5	1247 / 13279 / 1527	1730 / 18058 / 2039
5	8-9 / 10-5 / 6 -7	881 / 13925 / 1247	1138 / 18959 / 1730
6	10-11 / 12-7 / 8-9	1452 / 13720 / 881	1944 / 18745 / 1138
7	12-13 / 14-9 / 10-11	1589 / 13012 / 1452	2203 / 17680 / 1944
8	14-15 / 16-11 / 12-13	1434 / 13030 / 1589	1940 / 17684 / 2203
9	16-17 / 18-13 / 14-15	1480 / 13139 / 1434	2011 / 17876 / 1940
10	18-19 / 20-15 / 16-17	1241 / 13332 / 1480	1667 / 18149 / 2011
11	20-21 / 22-17 / 18-19	1151 / 13661 / 1241	1585 / 18575 / 1667
12	22-23 / 24-19 / 20-21	1291 / 13611 / 1151	1733 / 18509 / 1585

Table 8: Dataset statistics in cross-validation (Xval). Numbers are arranged in Dev/Train/Test order. Sections 6-1 denote sections 6-24 and sections 0-1.

Comparison.Concession	Comparison.Contrast
Contingency.Cause	Expansion.Conjunction
Expansion.Equivalence	Expansion.Instantiation
Expansion.Level-of-detail	Temporal.Asynchronous

Table 9: Second-level (L2) relations of PCC used in our experiments.

B Implementation Details

Table 10 shows the hyperparameter values for our model, most of which follow the default settings of RoBERTa (Liu et al., 2019). The value of temperature τ adopts from the default setting in Gumbel-Softmax. The k in inverse sigmoid decay is set to 100 for PDTB 2.0, 200 for PDTB 3.0, and 10 for PCC. We use different k for the three datasets because of their different sizes, and bigger datasets are assigned larger values. For a fair comparison, we equip baseline models with the same

Hyperparam	Value	Hyperparam	Value
Learning Rate	1e-5	Batch Size	16
Weigh Decay	0.1	Max Epochs	10
LR Decay	Linear	Warmup Ratio	0.06
Gradient Clipping	2.0	Max Seq Length	256
τ in Equation (4)	1.0	k in Equation (8)	100, 200, 10

Table 10: Hyperparameters for training our model.

	PDTB 2.0	
Models	Acc	F1
RoBERTaConn	96.69(+41.3)	95.58(+58.1)
Adversarial	74.93(+5.50)	68.62(+6.18)
Multi-Task	76.53(+5.71)	70.65(+6.86)
Pipeline	90.13(+19.1)	89.13(+24.5)
Our Model	83.71(+9.12)	79.25(+10.6)

Table 11: Level1 classification results on PDTB 2.0 (Ji split) when manually-annotated connectives are fed to connectives enhanced models. The numbers in brackets are performance gains compared to the default settings.

	PDTB 2.0	
Models	Acc	F1
Pipeline	58.15(-12.9)	46.68(-17.9)
Our Model	72.27(-2.32)	65.49(-3.15)

Table 12: Level1 classification results on PDTB 2.0 (Ji split) when generated connectives are removed from Pipeline and Our Method. The numbers in brackets are performance drops compared to the default settings.

$\text{RoBERTa}_{\text{base}}$[6][7] as our method and apply the same experimental settings (e.g. GPU, optimizer, learning rate, batch size, etc.) to them. For baselines that contain model-specific hyperparameters, such as the adversarial model (Qin et al., 2017), we follow their default setting described in the paper.

Considering the variability of training on PDTB, we report the mean performance of 5 random restarts for the "Ji" split (Ji and Eisenstein, 2015) and that of section-level cross-validation (Xval) like Kim et al. (2020). For PCC, we perform a 5-fold cross-validation on this corpus due to its limited number of data and report the mean results. We conduct all experiments on a single Tesla P40 GPU with 24GB memory. It takes about 110 minutes to train our model on every fold of PDTB 2.0, 150 minutes on every fold of PDTB 3.0, and 5 minutes on every fold of PCC.

For evaluation, we follow previous work (Ji and Eisenstein, 2015) to use accuracy (Acc, %) and F1-macro (F1, %) as metrics in our experiments.

[6]English version of RoBERTa-Base: https://huggingface.co/roberta-base

[7]German version: https://huggingface.co/benjamin/roberta-base-wechsel-german

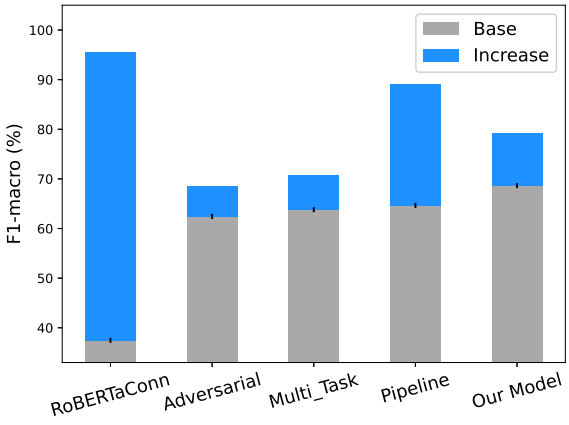

Figure 4: Level1 classification results (F1) on PDTB 2.0 (Ji split) when annotated connectives are fed to models. "Increase" denotes performance gain compared to the model with default settings ("Base").

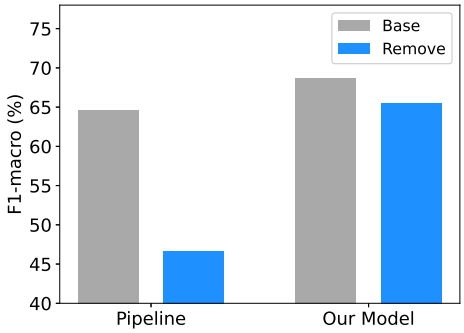

Figure 5: Level1 classification results (F1) on PDTB 2.0 (Ji split). "Remove" denotes the generated connectives are removed from the original model ("Base").

C Performance Analysis

Table 11 shows the Level1 classification results on PDTB 2.0 (Ji split) when manually-annotated connectives are fed to connective-enhanced models. Note that for models that do not use generated connectives, we insert the true connectives into their input in this setup. We also show the F1 results in Figure 4.

Table 12 shows the Level1 classification results on PDTB 2.0 when the generated connectives are removed from the inputs of relation classifiers in Pipeline and Our Method. This setting is not applied to other baselines, such as Multi-Task, because they either don't generate connectives or don't input the generated connectives into the relation classifiers. We also show the F1 results in Figure 5.

We investigate relation classifiers' performance of Multi-Task, Pipeline, and Our Model when connectives are correctly and incorrectly generated

Text	Label	RoBERTa	Adversarial	Multi-Task		Pipeline		Our Model	
		Rel	Rel	Conn	Rel	Conn	Rel	Conn	Rel
[I was trying to help kids in an unfair testing situation.]$_{Arg1}$ (Implicit=**Because**) [Only five of the 40 questions were geography questions.]$_{Arg2}$	Contingency	Expansion	Contingency	Because	Contingency	Because	Contingency	Because	Contingency
[The HUD budget has dropped by more than 70% since 1980.]$_{Arg1}$ (Implicit=**So**) [We've taken more than our fair share.]$_{Arg2}$	Contingency	Expansion	Expansion	So	Expansion	So	Contingency	So	Contingency
[In his lawsuit, Mr. Trudeau says the strike illegally included Darkhorse.]$_{Arg1}$ (Implicit=**In Response**) [A spokesman for the guild said the union's lawyers are reviewing the suit.]$_{Arg2}$	Expansion	Comparison	Comparison	However	Expansion	However	Comparison	However	Expansion
[Japan's swelling investment in Southeast Asia is part of its economic evolution.]$_{Arg1}$ (Implicit=**In particular**) [In the past decade, Japanese manufacturers concentrated on domestic production for export.. In the 1990s, spurred by rising labor costs and the strong yen, these companies will increasingly turn themselves into multinationals with plants around the world.]$_{Arg2}$	Expansion	Contingency	Expansion	Specifically	Expansion	Specifically	Expansion	Specifically	Expansion
[San Francisco Giants owner Bob Lurie hopes to have a new home for them.]$_{Arg1}$ (Implicit=**So**) [He is an avid fan of a proposition on next week's ballot to help build a replacement for Candlestick Park.]$_{Arg2}$	Contingency	Contingency	Expansion	In fact	Expansion	For example	Expansion	And	Expansion

Figure 6: Examples on 11-way prediction of PDTB 2.0 from different models. RoBERTa and Adversarial can only predict relations, while Multi-Task, Pipeline, and Our Model make predictions on both connectives and relations. Therefore, we show both Connective (Conn) and relation (Rel) prediction results of the latter three models. "Text" denotes input arguments and the annotated implicit connective, and "Label" means ground truth implicit relations. Correct predictions are marked in gray background.

(or predicted). Other baselines, such as Adversarial, are not included in this analysis because they don't predict or generate connectives. We mentioned in Section 4.3 that Multi-task, Pipeline, and Our Model's prediction on connective are different. Specifically, their predictions do not overlap and show different performances, with a mean accuracy of 31.30%, 33.21%, and 32.83% for Multi-Task, Pipeline, and Our Model, on PDTB 2.0, respectively.

Here, we show both good and bad cases of all models from correct and incorrect connective prediction groups in Figure 6. For comparison, we also show results from the RoBERTa and Adversarial baselines. In the first example, connective enhanced models, including Adversarial, Multi-Task, Pipeline, and Our Model, make the correct prediction on implicit relation with the help of connective information, while the RoBERTa baseline gives the wrong prediction. In the second example, Multi-Task, Pipeline, and Our Model all make the correct prediction on connectives. However, only the latter two correctly predict the implicit relations. We speculate this is because treating connectives as training objectives can not make full use of connectives. In the third example, all three models incorrectly predict the connective as "However". As a result, Pipeline incorrectly predicts the relation as "Comparison" due to the connective "However". Compared to it, both Multi-Task and Our Model correctly predict the relation "Expansion", showing better robustness. In the fourth example, all three models predict the connective as "Specifically", which is wrong but semantically similar to the manually-annotated connective "In particular". Consequently, those models all correctly predict the relation as "Expansion". In the final example, Multi-Task, Pipeline, and Our Model wrongly predict the connective as "In fact", "For example", and "And", respectively. And all three models are misled by the incorrect connectives, predicting the relation as "Expansion".

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
In Section 6.

☒ A2. Did you discuss any potential risks of your work?
Our paper is an entirely technical work. We don't think it has any risk of bias or otherwise.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
In Abstract section and Section 1.

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

B ☑ Did you use or create scientific artifacts?

In Section 4.1.

☑ B1. Did you cite the creators of artifacts you used?
In Section 4.1.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The datasets and tools we use are allowed for research purposes. For example, we are the member of LDC, so we can use the PDTB dataset for research.

☒ 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)?
Because the artifacts we used are produced for research purpose.

☒ 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?
Those corpora are extracted from news domain, and have been widely used in the field for a long time. We don't think it contains any offensive content.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
In Section 4.1

☑ 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.
In Appendix A

C ☑ Did you run computational experiments?

In Section 4.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
In Section 4.1 and Appendix B.

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
In Appendix B.

☑ 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?
In Section 4.2, 4.3, 4.4.

☒ 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.)?
Left blank.

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.?
No response.

☐ 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)?
No response.

☐ 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?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.