One General Teacher for Multi-Data Multi-Task: A New Knowledge Distillation Framework for Discourse Relation Analysis

Congcong Jiang, *Student Member, IEEE*, Tieyun Qian, *Member, IEEE*, Bing Liu, *Fellow, IEEE*

***Abstract*—Automatically identifying the discourse relations can help many downstream NLP tasks such as reading comprehension and machine translation. It can be categorized into explicit and implicit discourse relation recognition (EDRR and IDRR). Due to the lack of connectives, IDRR remains to be a big challenge. A good number of methods have been developed to combine explicit data with implicit ones under the multi-task learning framework. However, the difference in linguistic property and class distribution makes it hard to directly optimize EDRR and IDRR with multi-task learning.**

In this paper, we take the first step to exploit the knowledge distillation (KD) technique for discourse relation analysis. Our target is to train *a focused single-data single-task student* with the help of *a general multi-data multi-task teacher*. Specifically, we first train one teacher for both the top and second level relation classification tasks with explicit and implicit data. We then transfer the feature embeddings and soft labels from the teacher network to the student network. Moreover, we develop an adaptive knowledge distillation module to reduce the number of hyper-parameters and also to stimulate the potential of the student on autonomous learning. Extensive experimental results on the popular PDTB dataset proves that our model achieves a new state-of-the-art performance. We also show the effectiveness of our proposed KD architecture through detailed analysis.

***Index Terms*—Discourse Relation Analysis, Knowledge Distillation**

I. Introduction

Discourse relation recognition (DRR) aims to identify the discourse relations that hold between two text spans [1], [2]. DRR is a crucial step for many downstream natural language processing tasks, such as reading comprehension, automatic summarization, and machine translation. It consists of explicit and implicit discourse relation recognition (termed as EDRR and IDRR), whose difference depends on whether the connectives like 'as' exist or not in the data.

IDRR is a much more important and challenging task than EDRR. Implicit relations outnumber the explicit ones in naturally occurring texts [3]. Meanwhile, EDRR has approached a performance ceiling (higher than 90% F1 score) [4] while the state-of-the-art IDRR methods [5]–[8] only get about 60% F1 on the PDTB dataset.

Congcong Jiang and Tieyun Qian are with School of Computer Science, Wuhan University, China. E-mail: jiangcc@whu.edu.cn, qty@whu.edu.cn, Corresponding author: Tieyun Qian. Bing Liu is with University of Illinois at Chicago, USA. E-mail: liub@uic.edu. An early version of this paper was accepted as a four-page poster paper by the Web conference 2022 which is available at: https://www2022.thewebconf.org/PaperFiles/58.pdf

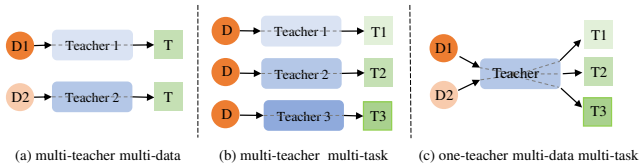


Fig. 1. The comparison among (a) multi-teacher multi-data, (b) multi-teacher multi-task, and (c) our one-teacher multi-data multi-task models. D and T denote data and task.

The success of EDRR can be largely owed to the existence and utilization of explicit connectives. With this in mind, researches on IDRR have made great efforts [3], [8]–[10] towards combining explicit data with implicit data under the multi-task learning (MTL) framework. However, *the problems of linguistic dissimilarity and different class distributions* [9], [11] make it hard to get optimal performance by directly training EDRR and IDRR with MTL. Recent advances in knowledge distillation (KD) [12]–[14] have proven that the knowledge can be transferred from a large teacher model to a small student model. Inspired by these pioneering studies, we propose to introduce the KD technique into IDRR for the first time. Our goal is to *retain the benefit of MTL in acquiring the common knowledge* across data or tasks, and to *exploit KD's power to transfer knowledge* from a multi-data multi-task teacher to a single-data single-task student.

Under the cross-data, cross-modal, or multiple languages circumstances, conventional KD methods [15]–[17] employ a multi-teacher multi-task or multi-data (denoted as MTMT/D) framework, where one teacher is trained for one task/data, as shown in Fig. 1 (a) (b). The MTMT/D architecture is effective for widely differing tasks since one single teacher is unable to grasp all knowledge. This resembles course teaching in primary school. For example, we need one teacher for math and another for English. Nevertheless, MTMT/D incurs a high computational cost since each teacher involves one neural network. More importantly, the cooperation among multiple teachers is also difficult. That's why the follow-up researches in this direction [18]–[20] have centered on the ensemble of teacher models.

In this paper, we argue that *MTMT/D is not always necessary* when the difference between tasks/data is not significant, e.g., geometry and algebra. The implicit and explicit data in our study also belong to this case. Moreover, the classification tasks in DRR often have connections. For example, in the commonly used PDTB [21] dataset, DRR has two levels

of relation classification task, corresponding to 4 top-level relations like 'Temporal' and 11 second-level relations like 'Temporal.Synchrony' and 'Temporal.Asynchronous'.

Based on the above analysis, we develop a novel *one-teacher multi-data multi-task (OTMT for short) knowledge distillation framework* for the IDRR task. As shown in Fig. 1 (c), OTMT trains a general teacher network for both the top and the second classification tasks with implicit and explicit data. (1) From the data perspective, one general teacher trained on different data can enhance the model's adaptivity to data, and thus the linguistic gap between implicit and explicit data can be naturally closed. (2) From the task perspective, one general teacher trained for different tasks can enforce the model to learn the connections, including the shared parameter space and the public features among tasks. When the model converges for all optimization objectives, it is empowered with the ability of generalization across tasks. (3) From the complexity perspective, one general teacher shares the data and the encoder structure in the same parameter space. As a result, it has the benefit of a small model size and also avoids the complicated ensemble procedure of multiple teacher models.

After training the general teacher network, its outputs are passed as training signals to supervise two separate student networks. Note that the student model focuses on the top/second classification task and is trained with implicit data only to alleviate the potential hurt caused by the mixture of explicit data. Furthermore, to reduce the number of hyper-parameters and also to stimulate the student' potential on autonomous learning, i.e., the student has the power to regulate the learning activities based on the sample and the teacher's quality, we develop an adaptive knowledge distillation module to further improve our proposed model and release the burden of hyper-parameter tuning. The experimental results on PDTB show that our proposed model achieves a new-state-of-the-art performance.

In summary, our contributions are as follows.

(1) We overcome the problems of existing multi-task learning (MTL) methods by introducing the KD technique, where a small student model can focus on IDRR, and meanwhile, it obtains the knowledge from a large teacher model trained on EDRR and IDRR.

(2) We design a novel one-teacher multi-task multi-data (OTMT) KD framework, where one general teacher can be efficiently trained, with the capability of better exploiting the common knowledge in related data and tasks.

(3) Extensive experiments verify the superior performance of our OTMT KD model over MTL methods, as well as the MTMT/D KD ones.

II. Related Work

DRR has received much attention since the release of PDTB 2.0 [21]. It contains two sub-tasks EDRR and IDRR. EDRR has achieved the human-level performance since its early stage [4], and thus we only review the literature in IDRR.

A. Implicit Discourse Relation Recognition

Feature Engineering. Some studies have focused on direct classification based on the observed arguments using feature engineering techniques [2], [22]–[26]. Other researches tend to discover semantic interactions between word pairs [27]–[31] or argument pairs [32]–[34] using deep learning methods.

Pre-trained Language Models (PLMs). With the success of PLMs, several fine-tuning or post-training methods [6], [35]–[37] are proposed for IDRR, with various types of PLMs engaged [5], [7], [38]–[42]. More recently, the large language models inspire new directions towards the generation based [43], [44] and prompt based methods [42], [45]. For example, Wu et al. [43] make the first attempt to deal with this task by the means of sequence generation. The seminal work [45] utilizes the prompt learning paradigm for IDRR. We also adopt the same PLMs as the encoders for a fair comparison with existing methods.

Multi Task Learning (MTL). To leverage the connectives in explicit data, a large number of methods [46]–[51] have been developed, either by turning explicit examples into pseudo implicit ones or by training the model under the multi-task framework. Another line of research is to train multiple IDRR-related tasks simultaneously [52]–[54]. Though effective, the linguistic dissimilarity between explicit and implicit data may hurt the performance [11]. Moreover, almost all previous studies perform one of the top or second level classification tasks [27]–[34] except the one in [54]. Hence the relation between the top and second level labels is largely neglected.

Different from the MTL framework in exploring multiple data or multi-level label relation, we introduce KD into this field, which can provide more informative soft labels and feature embeddings. To our knowledge, none of existing methods has exploited KD for DRR tasks before.

B. Knowledge Distillation (KD)

KD [12], [13] has been proven to be an effective way to explore soft labels. It starts from a single-teacher and then develops into multi-teacher architectures. Some work trains multiple teachers with the same data/task yet different networks [18]–[20], [55]–[57]. Some other trains them using different data [15], [16] or different tasks [17], [58], [59]. There is also a contemporaneous study [60] employing a meta-teacher network for the language model compression task across domains.

We differ our framework with prior knowledge distillation architectures in that this is the first attempt to training one general teacher in both the multi-data and multi-task circumstances.

III. Methodology

A. Problem Definition and Model Overview

Problem Definition Given a discourse argument pair $A = \{a_1, a_2, ..., a_{l1}\}$ and $B = \{b_1, b_2, ..., b_{l2}\}$, where $l1$ and $l2$ denote the number of words in A and B, DRR aims to predict the discourse relation between them. There is a relation hierarchy, including 4 top-level and 11 second-level relations. Correspondingly, there are two relation classification tasks. Moreover, depending on whether the connectives like 'because' exist in the argument pair, the data can be categorized into

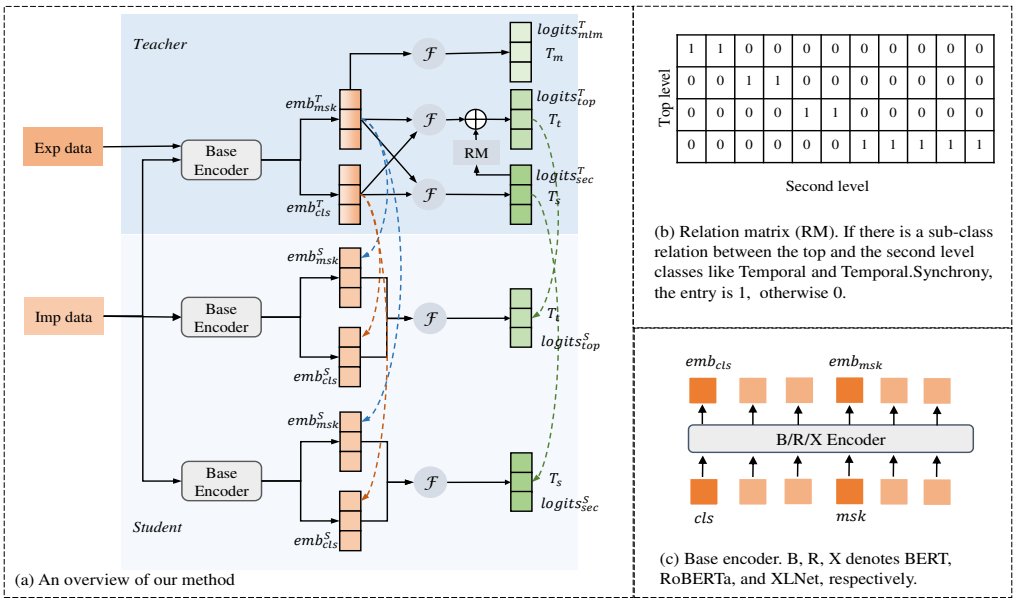


Fig. 2. The model architecture: (a) the model overview, (b) the relation matrix, i.e., the transfer matrix, and (c) the base encoder. The blue dashed line indicates the *teacher* $\Rightarrow$ *student* knowledge transfer. $\mathcal{F}$ denotes a fully connected layer.

explicit and implicit ones. DRR on these two types of data is termed as EDRR and IDRR, respectively.

The connectives are strong indicators of discourse relations and EDRR has been well addressed. In this study, we are interested in both the top and second level relation classification tasks on implicit data. Inspired by previous studies [46], [48], [50], we also deploy the explicit data to assist IDRR.

Model Overview The architecture of our proposed one general-teacher multi-data multi-task (OTMT) model is shown in Figure 2. It consist of one teacher network and two separate student networks.

The teacher and student networks in Fig. 2 (a) form the main block in the model's architecture for the knowledge distillation purpose. The relation matrix in Fig. 2 (b) is used to capture the sub-class relation between the top and the second level classes, such that these two-level tasks can help each other. The base encoder in Fig. 2 (c) is used to encode the sentence into the vector representation. Typical pre-trained language models including BERT, RoBERTa, and XLNet are adopted as the base encoder for a fair comparison with previous studies.

The teacher network is large and general. It takes both explicit and implicit data as input and performs three tasks: the masked language modeling task (T_m), the top level classification task (T_t), and the second level classification task (T_s). Two student networks are small and specific. They take the implicit data as input only and deal with a single task, i.e., one is for T_t and the other for T_s. For simplicity, we draw one of them in Fig. 2.

Base Encoder Following previous studies [5], [7], [36], we adopt several PLMs including BERT [61], RoBERTa [62], and XLNet [63] as the base encoder for both the teacher and student networks. PLMs have some special symbols: `[MASK]` is used to replace the masked tokens. `[CLS]` denotes the overall representation for the entire sentence. `[SEP]` is used to separate sentences. Each argument pair is input into the base encoder as follows:

$$[\texttt{CLS}], a_1, ..., a_{l1}, [\texttt{SEP}], [\texttt{MASK}], b_1, ..., b_{l2}, [\texttt{SEP}]. \quad (1)$$

B. Teacher Network for Multi-Data Multi-Task

In this section, we present our general teacher network for multi-data and multi-task. In particular, we use explicit and implicit data, and perform the top level and second level relation classification and an additional auxiliary masked language modeling (MLM) task to train the teacher network. By doing this, the teacher is equipped with deep and wide knowledge: it not only encodes the connections among different tasks, but also adapts to various types of data.

Multi-Data Our ultimate target is to train a good student for IDRR, thus the teacher network itself should also have the ability of handling implicit data. Moreover, by "seeing" explicit data, the teacher learns more knowledge about the relation sense between two arguments and also alleviates the data sparsity problem. That's why we take both explicit and implicit data as input to the teacher.

Connectives are strong signs for discourse relations. To exploit the connectives in explicit samples, we propose to align the explicit data with implicit ones. Specifically, we replace the connectives in explicit data with one or multiple `[MASK]` depending on whether the explicit connective contains one or multiple tokens. Meanwhile, we supplement a `[MASK]` for each implicit argument pair. The alignment operation enforces the `[MASK]` symbol to learn the semantic of connectives.

Both explicit and implicit data are sent into the base encoder, as shown in Eq. 1. We then get the vector representation of the input argument pair:

$$\boldsymbol{t}_{[\texttt{CLS}]}, \boldsymbol{t}^a_1, ..., \boldsymbol{t}^a_{l1}, \boldsymbol{t}_{[\texttt{SEP}]}, \boldsymbol{t}_{[\texttt{MASK}]}, \boldsymbol{t}^b_1, ..., \boldsymbol{t}^b_{l2}, \boldsymbol{t}_{[\texttt{SEP}]}, \quad (2)$$

where $\boldsymbol{t}$ denotes the token embedding in the teacher network.

Multi-Task The teacher network performs both the top and second level relation classification T_t and T_s, such that the model not only learns the task connection but also enhances the generalization ability across tasks. Furthermore, recall that we have aligned the explicit and implicit data. To better explore the real connectives in explicit data, we add another connective prediction task T_m to recover the masked connective in the teacher network. In summary, the teacher model performs three related tasks, i.e., T_t, T_s, and T_m.

We use the representation $\boldsymbol{t}_{[\texttt{CLS}]}$ and $\boldsymbol{t}_{[\texttt{MASK}]}$ for T_t and T_s since $\boldsymbol{t}_{[\texttt{CLS}]}$ encodes the information for the entire argument pair and $\boldsymbol{t}_{[\texttt{MASK}]}$ encodes the information for connectives. We also use $\boldsymbol{t}_{[\texttt{MASK}]}$ in explicit samples for T_m since the connective is actually known for explicit data. Formally, we have:

$$\begin{aligned}\boldsymbol{p}^T_{top,m} &= \text{softmax}(\mathbf{W}^T_{top,m}\boldsymbol{t}_{[\texttt{MASK}]} + \boldsymbol{b}^T_{top,m}),\\ \boldsymbol{p}^T_{top,c} &= \text{softmax}(\mathbf{W}^T_{top,c}\boldsymbol{t}_{[\texttt{CLS}]} + \boldsymbol{b}^T_{top,c}),\end{aligned} \tag{3}$$

$$\begin{aligned}\boldsymbol{p}^T_{sec,m} &= \text{softmax}(\mathbf{W}^T_{sec,m}\boldsymbol{t}_{[\texttt{MASK}]} + \boldsymbol{b}^T_{sec,m}),\\ \boldsymbol{p}^T_{sec,c} &= \text{softmax}(\mathbf{W}^T_{sec,c}\boldsymbol{t}_{[\texttt{CLS}]} + \boldsymbol{b}^T_{sec,c}),\end{aligned} \tag{4}$$

$$\boldsymbol{p}^T_{mlm} = \text{softmax}(\mathbf{W}^T_{mlm}\boldsymbol{t}_{[\texttt{MASK}]} + \boldsymbol{b}^T_{mlm}), \tag{5}$$

where $\boldsymbol{p}$ is the prediction logits, $\mathbf{W}$ and $\boldsymbol{b}$ are weight matrix and bias. top, sec, and mlm denote the top and second level classification task, and the masked connective prediction task. c and m denote using $\boldsymbol{t}_{[\texttt{CLS}]}$ and $\boldsymbol{t}_{[\texttt{MASK}]}$ for relation classification. The superscript T denotes the teacher network.

The top level discourse relations are course-grained, and each top level relation corresponds to two or more second level ones. For example, the 'Temporal' relation contains two second level relations, 'Temporal.Asynchronous' and 'Temporal.Synchrony'. The classification task usually becomes harder when the number of categories increases, which also holds for T_s. To address this problem, we consider to make use of the relation between the top and the second level classes to guide T_s. We introduce a relation matrix (RM). If a relation belongs to the second level class, it must belongs to the corresponding top level class too, and the entry in RM is 1 otherwise 0, as shown in Fig. 2 (b). RM is used as the constrains of the prediction logits between T_t and T_s.

$$\begin{aligned}\boldsymbol{p}^T_{top,c} &= \gamma * \mathbf{W}_{RM}\boldsymbol{p}^T_{sec,c} + (1-\gamma) * \boldsymbol{p}^T_{top,c},\\ \boldsymbol{p}^T_{top,m} &= \gamma * \mathbf{W}_{RM}\boldsymbol{p}^T_{sec,m} + (1-\gamma) * \boldsymbol{p}^T_{top,m},\end{aligned} \tag{6}$$

where $\mathbf{W}_{RM}$ is the weight matrix for the relation matrix RM, and γ $(0 < \gamma < 1)$ is the coefficient.

Training Loss for Teacher Network We adopt the cross entropy loss between the predicted logits and ground truth labels for training each of three tasks in the teacher network.

$$\begin{aligned}\mathcal{L}^T_{top} &= -\sum\nolimits_{j=1}^{|I|/|E|}\sum_{i\in\{c,m\}} \boldsymbol{y}^T_{top,j} * log(\boldsymbol{p}^T_{top,i,j}),\\ \mathcal{L}^T_{sec} &= -\sum\nolimits_{j=1}^{|I|/|E|}\sum_{i\in\{c,m\}} \boldsymbol{y}^T_{sec,j} * log(\boldsymbol{p}^T_{sec,i,j}),\\ \mathcal{L}^T_{mlm} &= -\sum\nolimits_{j=1}^{|E|} \boldsymbol{y}^T_{mlm,j} * log(\boldsymbol{p}^T_{mlm,j}),\end{aligned} \tag{7}$$

where $\boldsymbol{y}$ is ground truth labels and $\boldsymbol{p}$ is the predicted logits. j is the index of training samples, I and E denote the set of training samples of implicit and explicit data, respectively. The final loss $\mathcal{L}^T$ for the teacher network is the linear combination of the losses on three tasks.

$$\mathcal{L}^T = (1-\lambda) * (\mathcal{L}^T_{top} + \mathcal{L}^T_{sec}) + \lambda * \mathcal{L}^T_{mlm}, \tag{8}$$

where λ $(0 < \lambda < 1)$ is the coefficient.

We train the teacher network and save the teacher model that performs the best on the validation set of the IDRR task for knowledge distillation.

C. Student Network for Single-Data Single-Task

As shown in Figure 2, the student model takes the implicit data as the only input, and trains one network for the top and the second level classification task T_t and T_s separately. This means the student network is for single-data single-task such that it can be much focused and specific. Also note that there is no connective prediction task T_m since there is no explicit data in the student at all.

The student model adopts the base encoder with the same structure and same size as that in teacher network. After that, we can get the vector representation of the input argument pair as follows:

$$\begin{aligned}&\boldsymbol{s}_{task,[\texttt{CLS}]}, \boldsymbol{s}^a_{task,1}, ..., \boldsymbol{s}^a_{task,l1}, s_{task,[\texttt{SEP}]},\\ &\boldsymbol{s}_{task,[\texttt{MASK}]}, \boldsymbol{s}^b_{task,1}, ..., \boldsymbol{s}^b_{task,l2}, \boldsymbol{s}_{task,[\texttt{SEP}]},\end{aligned} \tag{9}$$

where $\boldsymbol{s}$ denotes the token embedding in the student network. Note that two student networks have their own parameter space. For clarity, we use $task \in \{top, sec\}$ to denote either the top or the second level classification task in the student networks.

Similar to the teacher network, we use the representation $\boldsymbol{s}_{task,[\texttt{CLS}]}$ and $\boldsymbol{s}_{task,[\texttt{MASK}]}$ for the top and second level classification.

$$\begin{aligned}\boldsymbol{p}^S_{task,m} &= \text{softmax}(\mathbf{W}^S_{task,m}\boldsymbol{s}_{task,[\texttt{MASK}]} + \boldsymbol{b}^S_{task,m}),\\ \boldsymbol{p}^S_{task,c} &= \text{softmax}(\mathbf{W}^S_{task,c}\boldsymbol{s}_{task,[\texttt{CLS}]} + \boldsymbol{b}^S_{task,c}),\end{aligned} \tag{10}$$

where $\boldsymbol{p}$ is the prediction logits, $\mathbf{W}$ and $\boldsymbol{b}$ are weight matrix and bias. c and m denote using $\boldsymbol{s}_{task,[\texttt{CLS}]}$ and $\boldsymbol{s}_{task,[\texttt{MASK}]}$ for relation classification. The superscript S denotes the student network.

We also adopt the cross entropy loss between the predicted logits and ground truth labels for training two separate student networks, i.e.,

$$\mathcal{L}^S_{task,g} = -\sum\nolimits_{j=1}^{|I|}\sum_{i\in\{c,m\}} \boldsymbol{y}^S_{task,j} * log(\boldsymbol{p}^S_{task,i,j}), \tag{11}$$

where $\boldsymbol{y}$ is ground truth labels and $\boldsymbol{p}$ is the predicted logits for the classification tasks. The subscript g denotes that this is the loss for ground truth labels.

D. Distilling Knowledge from Teacher to Student

To effectively transfer knowledge from the general multi-data and multi-task teacher to the single-data and single-task student networks, we propose to exploit two types of information learned by the teacher model including the soft labels and the feature vectors.

Transferring Soft Labels The teacher network also performs the top and second level relation classification tasks during the training process. The prediction made by the saved teacher network represents the teacher's judgement on the samples and contains rich inter-class information. Hence we

transfer such knowledge in the form of soft labels to supervise the student network.

$$\mathcal{L}^S_{task,s} = \sum_{j=1}^{|I|} \sum_{i\in\{c,m\}} \frac{\boldsymbol{p}^T_{task,i,j}}{\tau_{task}} * log(\boldsymbol{p}^S_{task,i,j}), \quad (12)$$

where $\boldsymbol{p}^T$ and $\boldsymbol{p}^S$ are the predictions generated by the saved teacher network and the student network using the same instance. τ is the temperature commonly used in knowledge distillation. The subscript s denotes that this is the loss for soft labels.

Transferring Feature Vectors Different from the student network, the teacher takes both the explicit and implicit data as input, and these two types of data are trained in one general teacher network within a multi-task framework and the shared parameter space. Hence the feature vectors obtained from the teacher network are informative. Based on this, we transfer them to the student network in addition to the soft labels, which is done by letting the feature vectors from the student as close as possible to those from the teacher.

$$\mathcal{L}^S_{task,v} = \sum_{j=1}^{|I|} \sum_{i\in\{c,m\}} 1 - sim(\boldsymbol{s}_{task,i,j}, \boldsymbol{t}_{task,i,j}), \quad (13)$$

where sim is the cosine similarity function. $\boldsymbol{s}_{task,i}$ and $\boldsymbol{t}_{task,i}$ are feature vectors in student and teacher network, respectively. The subscript v denotes that this is the loss for feature vectors.

Training Loss for Student Network In order to train each student network, we need to optimize the prediction and knowledge distillation targets at the same time. Hence the overall loss for the student network is defined as follows:

$$\mathcal{L}^S_{task} = \mathcal{L}^S_{task,g} + \alpha * \mathcal{L}^S_{task,s} + \beta * \mathcal{L}^S_{task,v}, \quad (14)$$

where g, s, v denote the loss for ground truth labels, soft labels, and features vectors. α and β are hyper-parameters.

E. Adaptive Knowledge Distillation

It is hard for to reproduce the results of KD algorithms due to the parameter selection issues. In order to reduce the number of hyper-parameters in our OTMT framework and also enhance the student's ability on autonomous learning, we add an adaptive distillation module, named OTMT-A, as a variant of the OTMT method. Specifically, OTMT-A replaces α and β in Eq.14 with the distillation weight which is adaptively calculated for each sample. The distillation weight W_t is measured from two aspects: the importance of the sample and the similarity between the teacher and the student.

Intuitively, if the gap between the teacher's soft labels and real labels is large, the sample should be paid more attentions during the learning process. This is in accordance with the basic idea in traditional semi-supervised learning. We use the Euclidean distance between the soft labels predicted by teacher model and the real labels to measure the importance of the sample. It is calculated as follows:

$$W_{t-imp} = D_E(y^S_{task,j}, \boldsymbol{p}^T_{task,i,j}). \quad (15)$$

where D_E denotes the Euclidean Distance, $i \in \{c, m\}$ and j denotes one of the samples in IDRR task.

In addition, the student's ability in absorbing the teacher's knowledge differs. Even with the same teacher, different students have different behaviors. Normally, if the student can keep pace with the teacher, she can usually learn more from the teacher. Therefore, we use the cosine similarity to measure the agreement between the teacher's soft labels and the student's predicted distribution. The higher the similarity, the greater impact the teacher can make on the student.

$$W_{t-sim} = sim(\boldsymbol{p}^T_{task,i,j}, \boldsymbol{p}^S_{task,i,j}), \quad (16)$$

Finally, the model adaptive distillation ratio W_t is calculated as follows:

$$W_t = W_{t-imp} * W_{t-sim}. \quad (17)$$

Therefore, the loss function of OTMT-A is shown in Eq.18.

$$\mathcal{L}^S_{task} = \mathcal{L}^S_{task,g} + W_t * (\mathcal{L}^S_{task,s} + \mathcal{L}^S_{task,v}). \quad (18)$$

IV. Experiments

We demonstrate the effectiveness of our model by thoroughly comparing it with various baselines. We also conduct an in-depth analysis to verify our design on architecture and components.

A. Dataset and Settings

1) Dataset Statistics and Splits: We use the second version of the Penn Discourse Treebank [21], i.e., PDTB 2.0 dataset. The link is available at: https://github.com/cgpotts/pdtb2. It contains 2,312 Wall Street Journal articles and has three levels of senses: Level-1 *Class*, Level-2 *Type*, and Level-3 *Subtype*. Following the convention, we adopt the Ji split [25] and accuracy and macro-averaged F1 metrics for the top level task, and use Ji [25], Lin [2], and P&K [36] splits and report the accuracy scores for the second level task. The data statistics are shown in Table I. We can see that the class distributions in implicit and explicit data are quite different.

TABLE I
DISTRIBUTIONS OF THE TOP LEVEL AND THE SECOND LEVEL RELATIONS OF IMPLICIT RELATION FROM THE TRAINING SECTIONS USING THE JI SPLIT [25].

Top	Second	Tr. Samples (Imp/Exp)
Temporal	Asynchronous	555/1590
	Synchrony	204/1192
Contingency	Cause	3276/1700
	Pragmatic Cause	64/9
Comparison	Contrast	1607/2883
	Concession	184/1021
Expansion	Conjunction	2879/3994
	Instantiation	1102/223
	Restatement	2458/119
	Alternative	152/282
	List	338/211

TABLE II
SECTION SPLITS

Splits	Train	Dev	Test
Ji	section 02-20	section 00-01	section 21-21
Lin	section 02-21	section 22	section 23
P&K	section 02-22	section 00-01	section 23-24

Following the convention, we adopt the Ji split [25] and accuracy [1] and macro-averaged F1 [2] metrics for the top level task, and use Ji [25], Lin [2], and P&K [36] splits and accuracy metric for the second level task. The details of split sections are shown in Table II.

2) Settings: Regarding the hyper-parameters, α and β in Eq. 14 are set to $\{1.0, 0.5\}$ and $\{1.0, 1.0\}$ for the top and second level tasks. The hyper-parameter γ in Eq. 6 is set to 0.1. λ in Eq. 8 is set to 0.05 for the top and second level tasks. The rationale for parameter setting and the detailed hyper-parameter study can be seen in Section V-D.

We save the teacher network that performs the best on the validation set of implicit relation data. After fixing parameters of the teacher model, we generate the corresponding soft labels and feature vectors for implicit samples, and use them together with the ground truth labels to guide the student network training.

For each student network, we train it up to 30 epochs and take the best models on the validation set, and then take the results on the test data. We report the averaged results over 5 runs with random initialization. This is consistent with previous studies [5]. We report the averaged results over 5 runs with random initialization on a 24 GB NVIDIA RTX 3090 GPU. The codes are already publicly available in a GitHub repository.

B. Main Results

We choose three types of baselines: (1) M1∼M4 are traditional methods which utilize single data and perform single task; (2) M5∼M16 are methods using multiple data or multiple tasks; (3) M16∼M21 are methods based on various types of PLMs. To fully compare our model with existing PLM based methods, we employ BERT [61], RoBERTa [62], and XLNet [63] as the base encoder and add the trainable segment embeddings to RoBERTa as [7] do.

The results for baselines are taken from their original papers except the third type of baselines (M16∼M21). This is because the PLM based models should be compared under the same encoder, and M16∼M21 are the methods which adopt the PLM based models as the encoder. We reproduce these six methods under same settings for fair comparison and statistical significance tests. The main comparison results between our model and the baselines on the top and second level classification are shown in Table III, respectively.

We can see that our proposed OTMT model shows significant improvements over the state-of-the-art baselines, in terms of both accuracy and F1 metrics, on the top level relation classification task in Table III. It also consistently outperforms all baselines on the second level task with different splits in almost all cases, as shown in Table III. [3]

Compared with the traditional single data or single task methods (M1∼M4) based on semantic discourse representation or feature engineering, simply expanding data (M5∼M7) gets limited improvements for IDRR tasks. On the contrary, the classic multi-data or multi-task methods (M8∼M16) show more positive impacts over the single-data or single-task ones.

The PLM based models (M16∼M20) benefit a lot from the pre-trained language models and show performance gaps over the traditional methods. Moreover, the larger corpus results in more gains than the small one, e.g., X_l vs. X_b. To exclude the influence of PLMs, the comparison among PLM based models should be conducted between the counterparts, e.g., X_b vs. X_b and X_l vs. X_l. B_b is the most widely adopted PLM. The better performance of [36] and [6] over their counterpart B_b baselines can be due to the fact that they both deploy millions of extra data for post-training BERT. So we did not compared with them. We notice that our model is better than M20 in most cases but slightly worse than it in terms of F1 with RoBERTa as the encoder. This hints that the prompt learning can benefit more from larger language models. However, M20 is unable to recognize the second-level relations. This is because M20 cannot verbalize the connectives to predict unique second-level relations due to overlapped connectives.

The idea of knowledge distillation rather than multi-task framework achieves the obviously improvement. Compared with M19, OTMT significantly exceeds it on both acc and F1 in the top level and the second level relation classification corresponding to the same encoder, and the performance of our RoBERTa-based OTMT also significantly exceeds the best baseline based on RoBERTa-based M18 in all aspects.

From Table III, we also find that the performance of OTMT-A is close to that of OTMT. This shows that it is feasible to selectively determine the distillation weight and empower the student with the ability of autonomous learning which can reduce the number of hyper-parameters in the model while ensuring the performance of the student model.

V. Detailed Analysis

A. Architecture Comparison

We first compare the performance and the complexity among different architectures. MTL replaces the KD architecture in our OTMT with a multi-task learning one. MTMD/T convert our one-teacher KD framework into a multi-teacher KD one. For MTMD, each teacher is trained by one data and multiple tasks; for MTMT, each teacher is trained by one task and multiple data. All the methods use BERT-base as the encoder. The results are shown in Table IV. We have several important notes.

(1) Our model achieves significantly better performance ($p < 0.01$) than MTL. This clearly proves that the introduction of our KD successfully transfers the knowledge from the large teacher to the small student. Our model also outperforms MTMT/D, showing that our one-teacher KD architecture has better adaptivity and generalization ability across data and tasks than multi-teacher or multi-data KD ones.

(2) In terms of complexity, the running time of our model is slightly longer than that of MTL, but much shorter than that of MTMT/D. By sharing the encoder, MTL has less space cost. Similarly, one teacher in our OTMT only needs one encoder

[1] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

[2] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

[3] The accuracy increase of OTMT over M16(B_b) on the top task is not significant since it uses the extra human-annotated connectives of implicit data. The fair comparison results between OTMT and M17(B_b) are all significant.

TABLE III
RESULTS (ACC%, F1%) ON THE TOP LEVEL TASK AND ACCURACY SCORES ON THE SECOND LEVEL TASK. † AND ‡ DENOTE STATISTICALLY SIGNIFICANT IMPROVEMENTS OVER THE CORRESPONDING (E.G., X_b VS. X_b) BEST BASELINES (WITH *) AT P<0.05 AND P<0.01.

Model	Top		Second (Acc)		
	Acc	F1	Lin	Ji	P&K
M1 (Liu and Li, 2016) [32]	57.57	44.95	–	–	–
M2 (Lei et al., 2018) [26]	–	47.15	–	–	–
M3 (Guo et al., 2020) [30]	57.25	47.90	–	–	–
M4 (He et al., 2020) [33]	59.94	51.24	–	–	–
M5 (Rutherford and Xue, 2015) [46]	57.10	40.50	–	–	–
M6 (Xu et al., 2018) [48]	60.63	44.48	–	–	–
M7 (Zhou et al., 2020) [50]	–	51.16	–	–	–
M8 (Lan et al., 2013) [9]	–	44.64	–	–	–
M9 (Liu et al., 2016b) [3]	57.27	44.98	–	–	–
M10 (Lan et al., 2017) [64]	57.39	47.80	–	–	–
M11 (Dai and Huang, 2018) [10]	57.44	48.82	–	–	–
M12 (Dai and Huang, 2019) [65]	59.66	52.89	–	48.23	–
M13 (Zhang et al., 2021) [8]	–	53.11	–	–	–
M14 (Bai and Zhao, 2018) [52]	–	51.06	45.73	48.22	–
M15 (Nguyen et al., 2019) [53]	–	53.00	46.48	49.95	–
M16 (Wu et al., 2020) [54]	61.74	52.62	–	49.76	–
M16 (Wu et al., 2020) [54](B_b)*	66.12	57.42	52.13	52.43	52.72
M17 (Shi and Demberg, 2019) [35](B_b)	66.01	57.17	52.12	52.32	52.34
M19 (Kim et al., 2020) [5](B_b)	65.52	56.27	51.94	51.89	51.88
M20 (Xiang et al., 2022) [45](B_b)	64.13	54.01	–	–	–
OTMT (B_b)	**66.94**	**59.19**†	**54.15**†	53.65†	**53.67**†
OTMT-A (B_b)	66.88	58.29†	53.07	**53.86**†	53.65†
M19 (Kim et al., 2020) [5](B_l)*	68.30	60.61	54.36	56.23	55.12
OTMT (B_l)	**70.02**‡	61.35†	56.03†	**57.55**†	56.99†
OTMT-A (B_l)	69.85‡	**61.98**†	**57.02**‡	57.48†	**57.37**†
M18 (Liu et al., 2020) [7](R_b)	67.14	57.84	52.38	55.39	55.15
M20 (Xiang et al., 2022) [45](R_b)*	69.79	**62.65**†	–	–	–
OTMT (R_b)	70.54‡	62.27	**56.87**‡	**58.02**‡	57.17†
OTMT-A (R_b)	**70.76**‡	61.99	56.58‡	57.98‡	**57.46**†
M19 (Kim et al., 2020) [5](X_b)*	66.35	59.33	54.33	54.62	54.36
OTMT (X_b)	**68.89**‡	60.78†	**56.37**‡	**56.65**‡	56.95‡
OTMT-A (X_b)	68.74‡	**60.81**†	56.29‡	56.19‡	**56.99**‡
M19 (Kim et al., 2020) [5](X_l)*	69.52	63.58	57.44	59.51	58.21
OTMT (X_l)	72.34‡	**64.46**†	**61.62**‡	61.06†	61.56‡
OTMT-A (X_l)	**72.61**‡	64.06	61.36‡	**61.50**†	**61.95**‡

TABLE IV
RESULTS FOR ARCHITECTURE COMPARISON. h=$hour$, M=1×10^6. † AND ‡ DENOTE STATISTICALLY SIGNIFICANT DIFFERENCE BETWEEN THE VARIANTS AND OTMT AT P<0.05 AND P<0.01.

	Top		Second (Acc)			Complexity	
	Acc	F1	Lin	Ji	P&K	Time	Space
OTMT (B_b)	**66.94**	**59.19**	**54.15**	**53.65**	**53.67**	1.18h	222M
MTL	61.66‡	51.11‡	50.65‡	48.41‡	50.05‡	**1.11**h	**110**M
MTMD	66.12	58.00	52.64†	52.38	53.20	1.87h	332M
MTMT	65.43	56.76‡	52.17†	52.97	53.18	2.64h	394M

TABLE V
ABLATION RESULTS. w/o DENOTES REMOVING THE COMPONENT. MLL DENOTES MULTI-LEVEL LABEL RELATION.

	Top		Second (Acc)		
	Acc	F1	Lin	Ji	P&K
OTMT (B_b)	**66.94**	**59.19**	**54.15**	**53.65**	**53.67**
w/o explicit data	66.71	58.33	51.44	53.37	53.13
w/o implicit data	64.78	55.13	51.30	49.89	51.85
w/o MLM	66.67	57.20	52.51	53.45	52.39
w/o RM	66.37	56.80	52.38	52.80	53.20
w/o student	61.66	51.11	50.65	48.41	50.05
w/o teacher	65.49	55.45	51.07	52.61	52.17
w/o soft-labels	66.38	57.50	52.40	52.96	**53.67**
w/o feature vectors	66.37	57.71	52.01	52.78	52.61
w/o MLL	64.89	55.81	53.18	53.07	52.84
w/o MLL & KD	62.36	52.11	51.13	49.22	50.38

while teachers in MTMT/D require two or three encoders and thus incur more complexity.

Overall, compared with multi-task learning, our model obtains great performance gains with small complexity loss. Compared with multi-teacher KD, our one-teacher KD model not only saves computational resources but also produces better classification performance.

B. Ablation Study

We conduct the ablation study from four aspects: (1) the types of data used for training the teacher model, (2) the tasks used for training the teacher model, (3) the KD for the student model, and (4) the KD vs. MTL in separate tasks. BERT based ablation results are shown in Table V.

It is clear that the complete OTMT is the best for both the top and second level tasks, indicating that all components

Argument Pairs	**AP1**, labels: Exp. & Exp.Conj				**AP2**, labels: Exp. & Exp.Rest				**AP3**, labels: Exp. & Exp.Conj			
	Arg1: Some 64% of items offered at the Guterman auction were sold, at an average … **Arg2**: The rest were withdrawn for lack of acceptable bids				**Arg1**: different British and American accounting standards produce higher… **Arg2**: Under British rules, Blue Arrow was able to write off at once the $1.15 billion in good will arising from the …				**Arg1**: His hands sit farther apart on the keyboard.Seventh chords make you feel as though he may break into a (very slow) improvisatory riff **Arg2**: The chords modulate			
	Top		**Second**		**Top**		**Second**		**Top**		**Second**	
OT MT	T: Exp. T: Comp.	0.58 0.37	T: Comp.Cont T: Exp.Conj	0.71 0.19	T: Exp. T: Cont.	0.83 0.14	T: Exp.Rest T: Exp.Inst	0.57 0.27	T: Temp. T: Exp.	0.88 0.10	T: Temp.Syn T: Exp.Conj	0.53 0.03
	S: Exp. S: Comp	0.45 0.44	S: Exp.Conj S: Comp.Cont	0.37 0.36	S: Exp. S: Cont	0.77 0.19	S: Exp.Rest S: Cont.Caus	0.28 0.26	S: Temp. S: Exp.	0.52 0.31	S: Exp.Conj S: Exp.Rest	0.50 0.18
MTL	Comp. Exp.	0.72 0.19	Exp.Conj Comp.Cont	0.56 0.35	Exp. Cont.	0.84 0.09	Exp.Inst Exp.Rest	0.23 0.22	Temp. Exp.	0.45 0.09	Temp.Syn Exp.Conj	0.82 0.02
MT MD	T1: Exp. T1: Comp.	0.37 0.36	T1: Comp.Cont T1: Exp.Conj	0.92 0.01	T1: Exp. T1: Cont.	0.43 0.12	T1: Exp.Conj T1: Exp.Rest	0.63 0.10	T1: Exp. T1: Comp	0.42 0.28	T1: Temp.Syn T1: Exp.Conj	0.85 0.12
	T2: Comp. T2: Exp.	0.45 0.43	T2: Exp.Conj T2: Comp.Cont	0.44 0.40	T2: Exp. T2: Cont.	0.78 0.16	T2: Exp.Rest T2: Exp.Conj	0.35 0.21	T2: Exp. T2: Temp.	0.83 0.06	T2: Exp.Conj T2: Exp.List	0.57 0.23
	S: Comp. S: Exp.	0.81 0.12	S: Comp.Cont S: Exp.Conj	0.52 0.22	S: Exp. S: Cont.	0.75 0.17	S: Exp.Inst S: Exp.Rest	0.30 0.25	S: Exp. S: Cont.	0.71 0.14	S: Exp.Conj S: Exp.Rest	0.23 0.16
MT MT	T1: Comp. T1: Exp.	0.58 0.03	T2: Exp.Conj T2: Comp.Cont	0.39 0.33	T1: Exp. T1: Cont.	0.98 0.01	T2: Exp.Inst T2: Exp.Rest	0.23 0.22	T1: Temp. T1: Exp.	0.43 0.36	T2: Exp.Conj T2: Comp.Cont	0.30 0.22
	S: Comp. S: Exp.	0.74 0.21	S: Comp.Cont S: Exp.Conj	0.48 0.34	S: Exp. S: Cont.	0.85 0.13	S: Cont.Caus S: Exp.Rest	0.38 0.24	S: Exp. S: Cont.	0.92 0.05	S: Exp.Conj S: Exp.Rest	0.83 0.06

Fig. 3. Results for case study. The top two prediction scores are presented for each method. The correct labels and their scores are underlined.

contribute to the model.

The removal of implicit data from the teacher model brings about more performance losses than that of explicit data. This is reasonable since the student cannot learn from the teacher about how to deal with the implicit data. In this case, it only obtains the relations between the top and the second level tasks.

We strengthen the connection between implicit and explicit data and that between the top and second level tasks via an additional masked connective prediction (MLM) task and the relation matrix (RM) constraint in the teacher model, respectively. We can find that removing either of them decreases the performance.

Among the KD components, '*w/o student*' denotes directly using the teacher model for IDRR, which incurs the biggest loss. '*w/o teacher*' denotes the student is trained only with the ground truth labels, which is also harmful. Compared with the soft label, we find that the removal of feature vectors has more negative impacts. All these results clearly prove the necessity and effectiveness of KD.

The effectiveness of our KD over the MTL framework can be further proved in two separate tasks. Firstly, our '*w/o explicit data*' variant has the similar setting as that of M16 (B_b) except that it utilizes the extra human-annotated connectives of implicit data which we don't. They exploit the MLL relation under KD and MTL, respectively, but our variant is better than M16 (B_b) in most circumstances. Secondly, our '*w/o MLL*' under KD significantly outperforms '*w/o MLL & KD*' under MTL. Note these two variants both employ explicit data. The results show that our KD is much better than MTL when exploiting either the MLL relation or explicit data.

C. Case Study

We show the results of three examples by different methods in Fig. 3 for a case study.

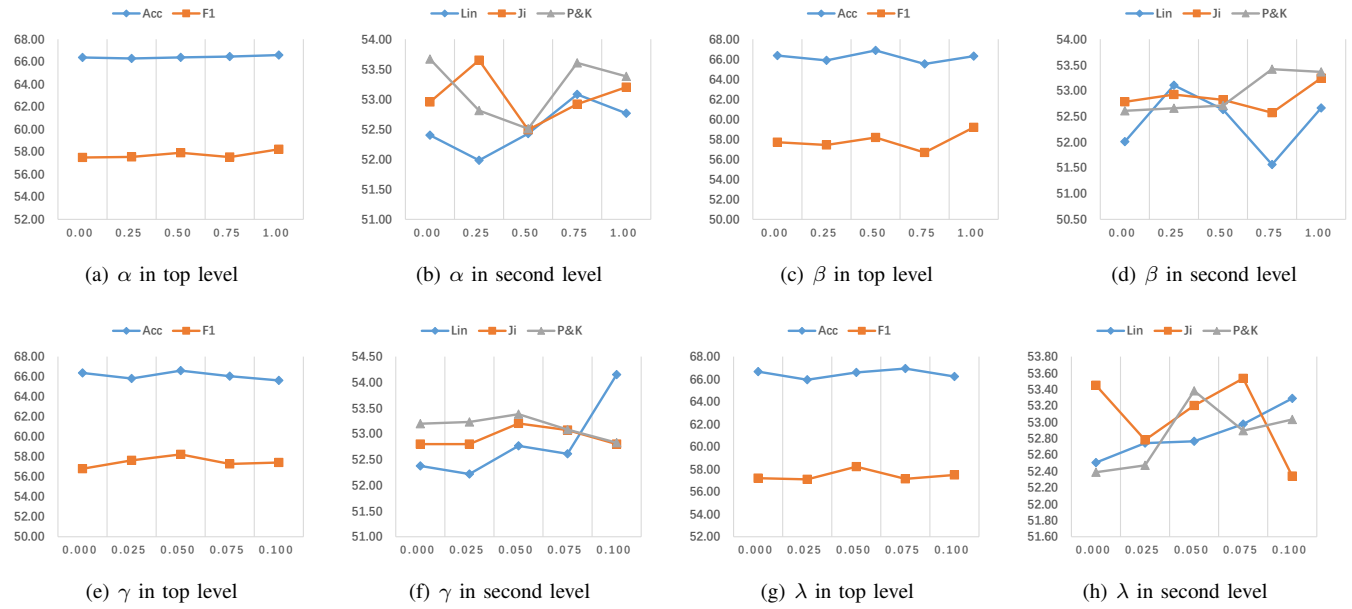
In AP1, on the top and second level tasks, MTMD has one correct teacher T2 for implicit data and one wrong teacher T1 for explicit data, but its student misses the correct labels on both tasks. This clearly reveals the difficulty to coordinate multiple teachers. Meanwhile, the correct label 'Exp.Conj' by T2 in MTMT for the second level task is impaired by its wrong label 'Comp' in T1 for the top level task. In contrast, both MTL and the trained student in our OTMT make correct predictions for the second level task.

In AP2, the top level classification is relatively easy, and all methods make correct predictions. However, none of the compared methods is right on the second level task. For example, since MTMT has separate teachers for the top and second level tasks, its T2 cannot benefit from the correct T1. Similarly, the results of T1 and T2 for different data in MTMD also conflict. On the contrary, the general teacher in our OTMT gets the common knowledge across data and tasks, and both the teacher and the student assign the correct label for the second level task.

Surely the single teacher model cannot always work better than multiple teachers as shown in AP3. When the only teacher predicts the wrong result and the wrong prediction probability is very high, it is easy to give the students wrong guidance. However, OTMT works better and is more efficient than MTMT and MTMD in most cases as we have shown in our paper.

D. Parameter Study

In this section, we present the parameter study to share our experience with the community and confidently reproduce the main results.



(a) α in top level (b) α in second level (c) β in top level (d) β in second level

(e) γ in top level (f) γ in second level (g) λ in top level (h) λ in second level

Fig. 4. Impacts of four hyper-parameters: α, β, γ, and λ.

TABLE VI
COMPARISON WITH DOMAIN ADAPTION METHODS.

	Top		Second (Acc)		
	Acc	**F1**	**Lin**	**Ji**	**P&K**
OTMT (B_b)	**66.94**	**59.19**	**54.15**	**53.65**	**53.67**
M22 (Ji et al., 2015) (B_b)	54.39	40.34	30.34	31.21	30.59
M22 (Ji et al., 2015) (B_b) + ImpData	63.82	54.55	51.47	50.19	51.24

The hyper-parameters α and β in Eq. 14 are used in the main training loss, and hence they are set using the grid search among $[0, 1]$ stepped by 0.25 for the top and second level tasks. The results are shown in Fig. 4 (a)$\sim$(d). Note the study for each hyper-parameter is carried out under other parameters' best settings. Also note these results are for the complete OTMT model. Our adaptive version OTMT-A does not contain these two hyper-parameters and thus there is no need to conduct this tuning procedure. As we can see from Fig. 4 (a)$\sim$(d), α and β have a small impact on the top level classification task, but have a relatively large impact on the second level task. This infers that the second level task is more sensitive to the supervision weights of the teacher model.

The hyper-parameter γ in Eq. 6 is used to determine the weight of the relation matrix RM, which serves as an auxiliary task, so γ should be set to a relatively small value. Similarly, the masked connective prediction task T_m is more difficult and its loss is much larger than the main T_t and T_s tasks. In order to prevent the model from over-satisfying T_m and neglecting T_t and T_s tasks, the coefficient λ in Eq. 8 should also be a small value. We hence roughly determine the reasonable value range for γ and λ. To this end, we compare the results when setting γ and λ to $\{0.001, 0.005, 0.01, 0.05, 0.1\}$. Through this, we conclude that both γ and λ should take a value between 0.0 and 0.1. After that, we conduct a grid search between these ranges stepped by 0.025 for γ and λ. The results for γ are shown in Fig. 4 (e) (f) and those for λ are shown in Fig. 4 (g) (h).

E. Comparison with Domain Adaption Methods

There are several cross-domain fusion methods [37], [47], [49] presented for unsupervised IDRR task. This problem is closely related to that in our work but the corresponding cross-domain fusion techniques are not suitable for our task. To verify this, we design two additional experiments. One is to refine the classic domain adaption method M22 [47] with BERT embeddings, and get a new M22 (B_b) version. We choose M22 [47] for comparison because the methods in [37], [49] have some complicated components which are tailored for their task. The other is to further add implicit data onto M22 (B_b). This ensures that the settings for M22 (B_b) + ImpData are same as those for our method. The results are shown in Table VI.

As can be seen, the cross-domain fusion methods M22 (B_b) and M22 (B_b) + ImpData for utilizing explicit data are significantly worse than our OTMT (B_b), and thus such methods are not suitable for our task.

VI. CONCLUSION

In this paper, we study the problem of implicit discourse relation recognition. To fully exploit the connectives in the explicit data and the relations between the hierarchical classification tasks, we propose a novel one-teacher multi-data multi-task KD framework. Better than multi-task learning, our

model leverages the KD's ability of transferring knowledge from a general teacher model to a specific student model. Different from multi-teacher KD, our model shares the common knowledge across multiple data and multiple tasks using one-teacher network with the low computational cost. Extensive experimental results on the popular PDTB dataset prove that our model significantly outperforms both the state-of-the-art baselines and the variants with the multi-task learning or multi-teacher KD architecture.

ACKNOWLEDGEMENTS

This work was supported by a grant from the National Natural Science Foundation of China (NSFC) project (No. 62276193). It was also supported in part by the Joint Laboratory on Credit Science and Technology of China Securities Credit Investment (CSCI), Wuhan University. The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

REFERENCES

[1] D. Marcu and A. Echihabi, "An unsupervised approach to recognizing discourse relations," in *ACL*, 2002, pp. 368–375.
[2] Z. Lin, M. Kan, and H. T. Ng, "Recognizing implicit discourse relations in the penn discourse treebank," in *EMNLP*, 2009, pp. 343–351.
[3] Y. Liu, S. Li, X. Zhang, and Z. Sui, "Implicit discourse relation classification via multi-task neural networks," in *AAAI*, 2016, pp. 2750–2756.
[4] E. Pitler and A. Nenkova, "Using syntax to disambiguate explicit discourse connectives in text," in *ACL*, 2009, pp. 13–16.
[5] N. Kim, S. Feng, R. C. Gunasekara, and L. A. Lastras, "Implicit discourse relation classification: We need to talk about evaluation," in *ACL*, 2020, pp. 5404–5414.
[6] Y. Kishimoto, Y. Murawaki, and S. Kurohashi, "Adapting BERT to implicit discourse relation classification with a focus on discourse connectives," in *LREC*, 2020, pp. 1152–1158.
[7] X. Liu, J. Ou, Y. Song, and X. Jiang, "On the importance of word and sentence representation learning in implicit discourse relation classification," in *IJCAI*, 2020, pp. 3830–3836.
[8] Y. Zhang, F. Meng, P. Li, P. Jian, and J. Zhou, "Context tracking network: Graph-based context modeling for implicit discourse relation recognition," in *NAACL-HLT*, 2021, pp. 1592–1599.
[9] M. Lan, Y. Xu, and Z. Niu, "Leveraging synthetic discourse data via multi-task learning for implicit discourse relation recognition," in *ACL*, 2013, pp. 476–485.
[10] Z. Dai and R. Huang, "Improving implicit discourse relation classification by modeling inter-dependencies of discourse units in a paragraph," in *NAACL-HLT*, 2018, pp. 141–151.
[11] C. Sporleder and A. Lascarides, "Using automatically labelled examples to classify rhetorical relations: an assessment," *Nat. Lang. Eng.*, vol. 14, no. 3, pp. 369–416, 2008.
[12] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning Workshop*, 2014.
[13] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *CVPR*, 2017, pp. 7130–7138.
[14] Z. Zhang, X. Shu, B. Yu, T. Liu, J. Zhao, Q. Li, and L. Guo, "Distilling knowledge from well-informed soft labels for neural relation extraction," in *AAAI*, 2020, pp. 9620–9627.
[15] S. Ruder, P. Ghaffari, and J. G. Breslin, "Knowledge adaptation: Teaching to adapt," *CoRR*, vol. abs/1702.02052, 2017.
[16] X. Tan, Y. Ren, D. He, T. Qin, Z. Zhao, and T. Liu, "Multilingual neural machine translation with knowledge distillation," in *ICLR*, 2019.
[17] K. Clark, M. Luong, U. Khandelwal, C. D. Manning, and Q. V. Le, "Bam! born-again multi-task networks for natural language understanding," in *ACL*, 2019, pp. 5931–5937.
[18] S. Du, S. You, X. Li, J. Wu, F. Wang, C. Qian, and C. Zhang, "Agree to disagree: Adaptive ensemble knowledge distillation in gradient space," in *NeurIPS*, 2020.
[19] Y. Liu, W. Zhang, and J. Wang, "Adaptive multi-teacher multi-level knowledge distillation," *Neurocomputing*, vol. 415, pp. 106–113, 2020.
[20] F. Yuan, L. Shou, J. Pei, W. Lin, M. Gong, Y. Fu, and D. Jiang, "Reinforced multi-teacher selection for knowledge distillation," in *AAAI*, 2021, pp. 14 284–14 291.
[21] R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. K. Joshi, and B. L. Webber, "The penn discourse treebank 2.0," in *LREC*, 2008.
[22] E. Pitler, A. Louis, and A. Nenkova, "Automatic sense prediction for implicit discourse relations in text," in *ACL*, 2009, pp. 683–691.
[23] Z. Zhou, Y. Xu, Z. Niu, M. Lan, J. Su, and C. L. Tan, "Predicting discourse connectives for implicit discourse relation recognition," in *COLING*, 2010, pp. 1507–1514.
[24] A. Rutherford and N. Xue, "Discovering implicit discourse relations through brown cluster pair representation and coreference patterns," in *EACL*, 2014, pp. 645–654.
[25] Y. Ji and J. Eisenstein, "One vector is not enough: Entity-augmented distributed semantics for discourse relations," *Trans. Assoc. Comput. Linguistics*, vol. 3, pp. 329–344, 2015.
[26] W. Lei, Y. Xiang, Y. Wang, Q. Zhong, M. Liu, and M. Kan, "Linguistic properties matter for implicit discourse relation recognition: Combining semantic interaction, topic continuity and attribution," in *AAAI*, 2018, pp. 4848–4855.
[27] J. Chen, Q. Zhang, P. Liu, X. Qiu, and X. Huang, "Implicit discourse relation detection via a deep architecture with gated relevance network," in *ACL*, 2016, pp. 1726–1735.
[28] W. Lei, X. Wang, M. Liu, I. Ilievski, X. He, and M. Kan, "SWIM: A simple word interaction model for implicit discourse relation recognition," in *IJCAI*, 2017, pp. 4026–4032.
[29] F. Guo, R. He, D. Jin, J. Dang, L. Wang, and X. Li, "Implicit discourse relation recognition using neural tensor network with interactive attention and sparse learning," in *COLING*, 2018, pp. 547–558.
[30] F. Guo, R. He, J. Dang, and J. Wang, "Working memory-driven neural networks with a novel knowledge enhancement paradigm for implicit discourse relation recognition," in *AAAI*, 2020, pp. 7822–7829.
[31] X. Li, Y. Hong, H. Ruan, and Z. Huang, "Using a penalty-based loss re-estimation method to improve implicit discourse relation classification," in *COLING*, 2020, pp. 1513–1518.
[32] Y. Liu and S. Li, "Recognizing implicit discourse relations via repeated reading: Neural networks with multi-level attention," in *EMNLP*, 2016, pp. 1224–1233.
[33] R. He, J. Wang, F. Guo, and Y. Han, "Transs-driven joint learning architecture for implicit discourse relation recognition," in *ACL*, 2020, pp. 139–148.
[34] H. Ruan, Y. Hong, Y. Xu, Z. Huang, G. Zhou, and M. Zhang, "Interactively-propagative attention learning for implicit discourse relation recognition," in *COLING*, 2020, pp. 3168–3178.
[35] W. Shi and V. Demberg, "Next sentence prediction helps implicit discourse relation classification within and across domains," in *EMNLP-IJCNLP*, 2019, pp. 5789–5795.
[36] A. Nie, E. Bennett, and N. D. Goodman, "Dissent: Learning sentence representations from explicit discourse relations," in *ACL*, 2019, pp. 4497–4510.
[37] M. Kurfali and R. Östling, "Let's be explicit about that: Distant supervision for implicit discourse relation classification via connective prediction," *CoRR*, vol. abs/2106.03192, 2021.
[38] W. Shi and V. Demberg, "Learning to explicitate connectives with seq2seq network for implicit discourse relation classification," in *IWCS*, 2019, pp. 188–199.
[39] Wei Shi and Vera Demberg, "Entity enhancement for implicit discourse relation classification in the biomedical domain," in *ACL/IJCNL*, 2021, pp. 925–931.
[40] C. Jiang, T. Qian, and B. Liu, "Knowledge distillation for discourse relation analysis. www (companion volume) 2022: 210-214," in *WWW (Companion Volume)*. ACM, 2022, pp. 210–215.
[41] Y. Jiang, L. Zhang, and W. Wang, "Global and local hierarchy-aware contrastive framework for implicit discourse relation recognition," in *Findings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.
[42] J. C. Z. L. Y. S. G. Y. W. S. S. Chunkit Chan, Xin Liu, "Discoprompt: Path prediction prompt tuning for implicit discourse relation recognition," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.
[43] C. Wu, L. Cao, Y. Ge, Y. Liu, M. Zhang, and J. Su, "A label dependence-aware sequence generation model for multi-level implicit discourse relation recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, pp. 11 486–11 494.

[44] W. Liu and M. Strube, "Annotation-inspired implicit discourse relation classification with auxiliary discourse connective generation," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.
[45] W. Xiang, Z. Wang, L. Dai, and B. Wang, "Connprompt: Connective-cloze prompt learning for implicit discourse relation recognition," in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 902–911.
[46] A. Rutherford and N. Xue, "Improving the inference of implicit discourse relations via classifying explicit discourse connectives," in *NAACL HLT*, 2015, pp. 799–808.
[47] Y. Ji, G. Zhang, and J. Eisenstein, "Closing the gap: Domain adaptation from explicit to implicit discourse relations," in *EMNLP*, 2015, pp. 2219–2224.
[48] Y. Xu, Y. Hong, H. Ruan, J. Yao, M. Zhang, and G. Zhou, "Using active learning to expand training data for implicit discourse relation recognition," in *EMNLP*, 2018, pp. 725–731.
[49] H. Huang and J. J. Li, "Unsupervised adversarial domain adaptation for implicit discourse relation classification," in *CoNLL*, 2019, pp. 686–695.
[50] M. Zhou, Q. Liang, L. Ma, D. Luo, P. Zhang, and B. Wang, "Towards selective data enhanced implicit discourse relation recognition via reinforcement learning," in *IJCNN*, 2020, pp. 1–8.
[51] C. Jiang, T. Qian, Z. Chen, K. Tang, S. Zhan, and T. Zhan, "Generating pseudo connectives with mlms for implicit discourse relation recognition," in *PRICAI*. Springer, 2021, pp. 113–126.
[52] H. Bai and H. Zhao, "Deep enhanced representation for implicit discourse relation recognition," in *COLING*, 2018, pp. 571–583.
[53] L. T. Nguyen, N. V. Linh, K. Than, and T. H. Nguyen, "Employing the correspondence of relations and connectives to identify implicit discourse relations via label embeddings," in *ACL*, 2019, pp. 4201–4207.
[54] C. Wu, C. Hu, R. Li, H. Lin, and J. Su, "Hierarchical multi-task learning with CRF for implicit discourse relation recognition," *Knowl. Based Syst.*, vol. 195, p. 105637, 2020.
[55] Y. Chebotar and A. Waters, "Distilling knowledge from ensembles of neural networks for speech recognition," in *Interspeech*, 2016, pp. 3439–3443.
[56] S. You, C. Xu, C. Xu, and D. Tao, "Learning from multiple teacher networks," in *SIGKDD*, 2017, pp. 1285–1294.
[57] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran, "Efficient knowledge distillation from an ensemble of teachers," in *Interspeech*, 2017, pp. 3697–3701.
[58] X. Liu, P. He, W. Chen, and J. Gao, "Improving multi-task deep neural networks via knowledge distillation for natural language understanding," *CoRR*, vol. abs/1904.09482, 2019.
[59] W. Li and H. Bilen, "Knowledge distillation for multi-task learning," in *ECCV*, vol. 12540, 2020, pp. 163–176.
[60] H. Pan, C. Wang, M. Qiu, Y. Zhang, Y. Li, and J. Huang, "Meta-kd: A meta knowledge distillation framework for language model compression across domains," in *ACL/IJCNLP*, 2021, pp. 3026–3036.
[61] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019, pp. 4171–4186.
[62] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.
[63] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *NeurIPS*, 2019, pp. 5754–5764.
[64] M. Lan, J. Wang, Y. Wu, Z. Niu, and H. Wang, "Multi-task attention-based neural networks for implicit discourse relationship representation and identification," in *EMNLP*, 2017, pp. 1299–1308.
[65] Z. Dai and R. Huang, "A regularization approach for incorporating event knowledge and coreference relations into neural discourse parsing," in *EMNLP-IJCNLP*, 2019, pp. 2974–2985.

Congcong Jiang received the BS and MS degree from School of Computer Science, Wuhan University, China in 2019 and 2022. Her research interests include natural language processing, dialogue system. She has authored or coauthored several papers in major conferences, such as the web conference and PRICAI.

Tieyun Qian is a professor at the School of Computer Science, Wuhan University, China. She received her BS degree in Computer Science from Wuhan University of Technology, China in 1991, and her Ph.D. degree in Computer Science from Huazhong University of Science and Technology, China in 2006. Her current research interests include text mining, Web mining, and natural language processing. She has published over 80 papers in leading conferences and top journals including ACL, AAAI, SIGIR, TKDE, TOIS, TASLP. She is a member of ACM, IEEE, and CCF. She has served as program committee member or area chair of many premium conferences like ACL, WWW, AAAI, and IJCAI.

Bing Liu (Fellow, IEEE) received the Ph.D. degree in artificial intelligence from The University of Edinburgh, Edinburgh, U.K. He is currently a Distinguished Professor of computer science with the University of Illinois at Chicago, Chicago, IL, USA. He has authored or coauthored extensively in top conferences and journals, and authored four books: one about lifelong learning, two about sentiment analysis, and one about web mining. His research interests include lifelong and continual learning, sentiment analysis, lifelong learning chat bots, open-world AI/learning, natural language processing, and data mining and machine learning. From 2013–2017, he was the Chair of ACM SIGKDD, a Program Chair of many leading data mining conferences. He is a Fellow of the ACM and AAAI.