Knowledge Distillation for Discourse Relation Analysis

Congcong Jiang
jiangcc@whu.edu.cn
School of Computer Science,
Wuhan University
China

Tieyun Qian
qty@whu.edu.cn
School of Computer Science,
Wuhan University
China

Bing Liu
liub@uic.edu
Department of Computer Science,
University of Illinois at Chicago
USA

ABSTRACT

Automatically identifying the discourse relations can help many downstream NLP tasks such as reading comprehension. 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. 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. 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.

CCS CONCEPTS

• **Computing methodologies** → **Discourse, dialogue and pragmatics**.

KEYWORDS

text mining, discourse relation analysis, knowledge distillation.

ACM Reference Format:
Congcong Jiang, Tieyun Qian, and Bing Liu. 2022. Knowledge Distillation for Discourse Relation Analysis. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3487553.3524245

1 INTRODUCTION

Discourse relation recognition (DRR) aims to identify the discourse relations that hold between two text spans. DRR is a crucial step for many downstream natural language processing tasks. 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. 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 towards combining explicit data with implicit data under the multi-task learning (MTL) framework. However, *the problems of linguistic dissimilarity and different class distributions* [4] make it hard to get optimal performance by directly training EDRR and IDRR with MTL. Recent advances in knowledge distillation (KD) [1] 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.

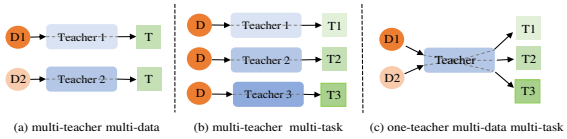

Figure 1: The comparison among (a) multi-teacher multi-data, (b) multi-teacher multi-task, and (c) our one-teacher multi-data multi-task models. D and T denote data and task.

Under the cross-data, cross-modal, or multiple languages circumstances, conventional KD methods 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 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 [8] 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'.

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

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524245

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. The experimental results on PDTB show that our proposed model achieves a new-state-of-the-art performance.

2 METHODOLOGY

2.1 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. There are two relation classification tasks (4 top-level and 11 second-level). DRR is categorized into EDRR and IDRR where EDRR has been well addressed. We are interested in both the top and second level relation classification tasks on implicit data, and 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 consists of one teacher network and two separate student networks. 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 [3, 6, 7], we adopt several Pre-trained Language Models (PLMs) including BERT, RoBERTa, and XLNet 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:

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

2.2 Teacher Network

We first present our general teacher network for multi-data and multi-task. 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.

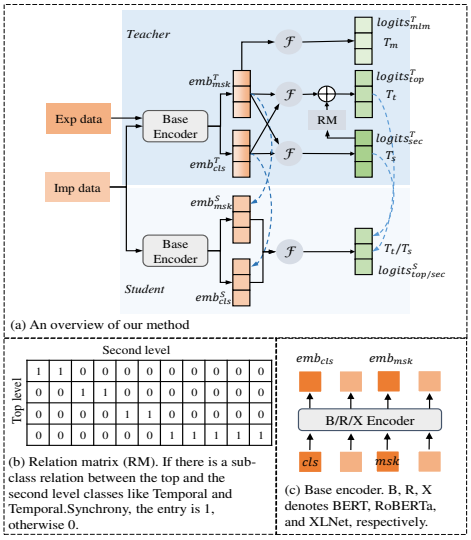

Figure 2: The model architecture: (a) model overview, (b) transfer matrix, and (c) base encoder. The blue dashed line indicates the *teacher* ⇒ *student* knowledge transfer. $\mathcal{F}$ denotes a fully connected layer.

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:

$$\boldsymbol{p}^T_{t|s,m|c} = \text{softmax}(\mathbf{W}^T_{t|s,m|c}\boldsymbol{t}_{\texttt{[MASK]}||\texttt{[CLS]}} + \boldsymbol{b}^T_{t|s,m|c}), \quad (3)$$

$$\boldsymbol{p}^T_{mlm} = \text{softmax}(\mathbf{W}^T_{mlm}\boldsymbol{t}_{\texttt{[MASK]}} + \boldsymbol{b}^T_{mlm}), \quad (4)$$

where $\boldsymbol{p}$ is the prediction logits, $\mathbf{W}$ and $\boldsymbol{b}$ are weight matrix and bias. t, s, 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. 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 belong 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.

$$\boldsymbol{p}^T_{t,c|m} = \gamma * \mathbf{W}_{RM}\boldsymbol{p}^T_{s,c|m} + (1-\gamma) * \boldsymbol{p}^T_{t,c|m}, \tag{5}$$

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.

$$\begin{aligned}\mathcal{L}^T_{t|s} &= -\sum_{j=1}^{|I|/|E|}\sum_{i\in\{c,m\}} \boldsymbol{y}^T_{t|s,j} * log(\boldsymbol{p}^T_{t|s,i,j}),\\ \mathcal{L}^T_{mlm} &= -\sum_{j=1}^{|E|} \boldsymbol{y}^T_{mlm,j} * log(\boldsymbol{p}^T_{mlm,j}),\end{aligned} \tag{6}$$

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.

$$\mathcal{L}^T = (1-\lambda) * (\mathcal{L}^T_t + \mathcal{L}^T_s) + \lambda * \mathcal{L}^T_{mlm}, \tag{7}$$

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.

2.3 Student Network

As shown in Fig. 2, the student model takes 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:

$$\boldsymbol{s}_{task,[\texttt{CLS}]}, \boldsymbol{s}^a_{task,1}, ..., \boldsymbol{s}_{task,[\texttt{SEP}]}, \boldsymbol{s}_{task,[\texttt{MASK}]}, \boldsymbol{s}^b_{task,1}, ..., \boldsymbol{s}_{task,[\texttt{SEP}]}, \tag{8}$$

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.

$$\boldsymbol{p}^S_{task,m|c} = \text{softmax}(\mathbf{W}^S_{task,m|c}\boldsymbol{s}_{task,[\texttt{MASK}]||[\texttt{CLS}]} + \boldsymbol{b}^S_{task,m|c}), \tag{9}$$

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 adopt the cross entropy loss between the predicted logits and ground truth labels for training two separate student networks:

$$\mathcal{L}^S_{task,g} = -\sum_{j=1}^{|I|}\sum_{i\in\{c,m\}} \boldsymbol{y}^S_{task,j} * log(\boldsymbol{p}^S_{task,i,j}), \tag{10}$$

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.

2.4 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.

$$\mathcal{L}^S_{task,s} = \sum_{j=1}^{|I|}\sum_{i\in\{c,m\}} \frac{\boldsymbol{p}^T_{task,i,j}}{\tau_{task}} * log(\boldsymbol{p}^S_{task,i,j}), \tag{11}$$

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.

$$\mathcal{L}^S_{task,v} = \sum_{j=1}^{|I|}\sum_{i\in\{c,m\}} 1 - sim(\boldsymbol{s}_{task,i,j}, \boldsymbol{t}_{task,i,j}), \tag{12}$$

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:

$$\mathcal{L}^S_{task} = \mathcal{L}^S_{task,g} + \alpha * \mathcal{L}^S_{task,s} + \beta * \mathcal{L}^S_{task,v}, \tag{13}$$

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

3 EXPERIMENTS

3.1 Settings

We adopt 4 top-level and 11 second-level relations as categories in PDTB 2.0 [8] for DRR, and use the Ji split [2] and accuracy and macro-averaged F1 metrics for the top level task, and use Ji [2],

Table 1: 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 [10](Bb)*	66.12	57.42	52.13	52.43	52.72
M2 [9](Bb)	66.01	57.17	52.12	52.32	52.34
M3 [6](Rb)*	67.14	57.84	52.38	55.39	55.15
M4 [3](Bb)	65.52	56.27	51.94	51.89	51.88
M4 [3](Bl)*	68.30	60.61	54.36	56.23	55.12
M4 [3](Xb)*	66.35	59.33	54.33	54.62	54.36
M4 [3](Xl)*	69.52	63.58	57.44	59.51	58.21
OTMT (Bb)	**66.94**	**59.19**†	**54.15**†	**53.65**†	**53.67**†
OTMT (Bl)	**70.02**‡	**61.35**†	**56.03**‡	**57.55**†	**56.99**†
OTMT (Rb)	**70.54**‡	**62.27**‡	**56.87**‡	**58.02**‡	**57.17**†
OTMT (Xb)	**68.89**‡	**60.78**†	**56.37**‡	**56.65**‡	**56.95**‡
OTMT (Xl)	**72.34**‡	**64.46**†	**61.62**‡	**61.06**†	**61.56**‡

Lin [5], and P&K [7] splits and report the accuracy scores for the second level task. To fully compare our model with existing PLM based methods, we employ BERT (B), RoBERTa (R), and XLNet (X) as the base encoder with both base (b) and large (l) models.

We save the teacher network that performs the best on the val. set of implicit relation data. We then 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, save the best model on the dev. set, and then take the results on the test data. We report the averaged results over 5 runs with random initialization. α and β in Eq. 13 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. 5 is set to 0.1. λ in Eq. 7 is set to 0.05.

3.2 Main Results

The state-of-the-art PLM based methods (M1~M4) are chosen as baselines since they have shown performance gaps over the traditional methods. We reproduce them using the same settings as OTMT. The comparison results are shown in Table 1. It is clear that our OTMT shows significant improvements over baselines, in terms of both accuracy and F1 metrics, on the top level relation classification task in Table 1. [1] It also consistently outperforms all baselines on the second level task with all different splits.

3.3 Detailed Analysis

We perform architecture and ablation study and show the results in Table 2. We first compare the performance and the complexity among different architectures. MTL replaces the KD architecture in OTMT with a multi-task learning one. MTMD/T converts our one-teacher KD framework into a multi-teacher KD one, where each teacher is trained by one data and multiple tasks/one task and multiple data. All methods are BERT-based and use same settings.

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

[1]The accuracy increase of OTMT over M1(Bb) on the top task is not significant since it uses the extra human-annotated connectives. The fair comparison results between OTMT and M2(Bb) are all significant.

Table 2: Results for architecture comparison (upper) and ablation study (lower). *h*=*hour*, M=1×10^6.

	Top		Second (Acc)			Complexity	
	Acc	F1	Lin	Ji	P&K	Time	Space
OTMT	**66.94**	**59.19**	**54.15**	**53.65**	**53.67**	1.18*h*	222*M*
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	1.87*h*	332*M*
MTMT	65.43	56.76‡	52.17†	52.97	53.18	2.64*h*	394*M*
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 label	66.38	57.50	52.40	52.96	53.67	-	-
w/o feature vector	66.37	57.71	52.01	52.78	52.61	-	-

Among the ablation results for KD components, '*w/o student*' denotes directly using the teacher model for IDRR, which deteroirates into MTL and incurs the biggest loss. '*w/o teacher*' denotes the student is trained only with the ground truth labels, which is also harmful. Since feature vectors are more direct signals than soft labels, '*w/o feature vector*' has more negative impacts than '*w/o soft label*' especially on the second level task.

4 CONCLUSION

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.

REFERENCES

[1] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2014. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning Workshop*.

[2] Yangfeng Ji and Jacob Eisenstein. 2015. One Vector is Not Enough: Entity-Augmented Distributed Semantics for Discourse Relations. *TACL* (2015).

[3] Najoung Kim, Song Feng, R. Chulaka Gunasekara, and Luis A. Lastras. 2020. Implicit Discourse Relation Classification: We Need to Talk about Evaluation. In *ACL*.

[4] Man Lan, Yu Xu, and Zheng-Yu Niu. 2013. Leveraging Synthetic Discourse Data via Multi-task Learning for Implicit Discourse Relation Recognition. In *ACL*.

[5] Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing Implicit Discourse Relations in the Penn Discourse Treebank. In *EMNLP*. 343–351.

[6] Xin Liu, Jiefu Ou, Yangqiu Song, and Xin Jiang. 2020. On the Importance of Word and Sentence Representation Learning in Implicit Discourse Relation Classification. In *IJCAI*. 3830–3836.

[7] Allen Nie, Erin Bennett, and Noah D. Goodman. 2019. DisSent: Learning Sentence Representations from Explicit Discourse Relations. In *ACL*. 4497–4510.

[8] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K. Joshi, and Bonnie L. Webber. 2008. The Penn Discourse TreeBank 2.0. In *LREC*.

[9] Wei Shi and Vera Demberg. 2019. Next Sentence Prediction helps Implicit Discourse Relation Classification within and across Domains. In *EMNLP-IJCNLP*.

[10] Changxing Wu, Chaowen Hu, Ruochen Li, Hongyu Lin, and Jinsong Su. 2020. Hierarchical multi-task learning with CRF for implicit discourse relation recognition. *Knowl. Based Syst.* 195 (2020), 105637.