Co-Teaching Student-Model through Submission Results of Shared Task

Kouta Nakayama[1,2] **Shuhei Kurita**[1] **Akio Kobayashi**[3]
Yukino Baba[2] **Satoshi Sekine**[1]
[1]RIKEN [2]University of Tsukuba
[3]National Agriculture and Food Research Organization
{kouta.nakayama, shuhei.kurita, satoshi.sekine}@riken.jp
akio.kobayashi@naro.go.jp, baba@cs.tsukuba.ac.jp

Abstract

Shared tasks have a long history and have become the mainstream of NLP research. Most of the shared tasks require participants to submit only system outputs and descriptions. It is uncommon for the shared task to request submission of the system itself because of the license issues and implementation differences. Therefore, many systems are abandoned without being used in real applications or contributing to better systems. In this research, we propose a scheme to utilize all those systems which participated in the shared tasks. We use all participated system outputs as task teachers in this scheme and develop a new model as a student aiming to learn the characteristics of each system. We call this scheme "Co-Teaching." This scheme creates a unified system that performs better than the task's single best system. It only requires the system outputs, and slightly extra effort is needed for the participants and organizers. We apply this scheme to the "SHINRA2019-JP" shared task, which has nine participants with various output accuracies, confirming that the unified system outperforms the best system. Moreover, the code used in our experiments has been released.[1]

1 Introduction

Shared tasks have a long history and have become the highlight of NLP research (Sundheim, 1995; Tjong Kim Sang and Buchholz, 2000; Ounis et al., 2008; Dang and Owczarzak, 2009). These tasks have contributed to natural language processing technology development by attracting researchers interested in being the best task player. The systems are evaluated using the output submitted to the task, and they usually have no obligation to submit the system. It limits the participant's contribution once the task is over because the system is a future asset. We believe all participating systems have values as a resource, even if they are not the best. It may be desirable to share it for the sake of innovation in the field as a whole. However, sharing the system is challenging because of the license issue and the running environment. Although sharing the system is ideal, we believe sharing system outputs are much easier, and only slight additional effort is needed for the task participants and organizers. We propose a scheme to utilize all system outputs in the shared task to build a unified system that is better than the best single system. More specifically, we construct a system that reproduces the participating systems embedded in the submission results by treating the system submission results as training data (teacher) and building a new model (student). Here, those submissions include evaluation data and large unlabeled data submissions. This is an adaptation of the Teacher-Student architecture of model compression methods such as knowledge distillation (Ba and Caruana, 2014; Hinton et al., 2015). We call this scheme "Co-Teaching," borrowing from real-world educational terminology, because the group of participants in the shared task act as teachers and teach a common student. This scheme can be applied to most shared tasks, as it requires submitting the evaluation data and the unlabeled data results. Its benefits include building a better system for the task, as well as salvaging the effort of the participants who did not produce the top results. The scheme can provide the best-performing system without violating the participant's license, and it is also possible to design a shared task so that it aims to build a single system from the beginning.

In order to prove the effectiveness of the proposed scheme, we conducted an experiment on the SHINRA2019-JP task. The task is to extract values corresponding to predefined attributes from Wikipedia articles to structure the Japanese Wikipedia. SHINRA2019-JP follows the con-

[1] https://github.com/k141303/co_teaching_scheme

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4525–4535

cept of "Resource by Collaborative Contribution (RbCC)" (Sekine et al., 2019), in which a resource (e.g., a knowledge base) is built within the framework of a shared task, and the submission results are made publicly available as a resource. The evaluation data is not announced for this task, and the participants are required to submit results for all Wikipedia articles. There are many system predictions for the unlabeled portion of the data due to this evaluation setting. We use ensemble learning to create better results for this task since the outputs are all public. The details of the SHINRA2019-JP task are described in Sec. 4.1.

"RbCC" is a scheme to create a resource collaboratively, but our proposed Co-Teaching scheme aims to create a system out of many submitted outputs. We trained the student model to build a system using publicly available submission results and the training data distributed to the task participants. The result shows that the proposed system achieves a better score than the best participating system.

The contributions of this paper can be summarized below:

- By proposing a Co-Teaching scheme, i.e., building a single system via a shared task, we have exhibited a new way of utilizing shared tasks. To the best of our knowledge, there is no effort to exploit the participant's effort by releasing the integrated system.
- We applied the proposed scheme to an actual shared task, SHINRA2019-JP, and demonstrated that the system proposed by the scheme achieves a better score than the best participating systems. Additionally, we proved the effectiveness of using the participant results in ablation tests.
- We enumerated the shared tasks that have been conducted recently in the field of natural language processing and discussed our scheme's applicability.

2 Related Work

2.1 Knowledge Distillation

Knowledge Distillation (Ba and Caruana, 2014; Hinton et al., 2015) is a method mainly used for model compression. Specifically, the results of a model with many parameters, or the ensemble results of multiple models, are used as training data to learn a new model with fewer parameters. In this case, the learning source model is called the teacher model, the learning destination is called the student model, and this combination is called the Teacher-Student architecture. The learning itself is called distillation. In many cases, the teacher model is supervised, and the same training data is also used when training student. In the Teacher-Student architecture, there are two categories of knowledge transfer methods: response-based (Ba and Caruana, 2014; Hinton et al., 2015) and feature-based (Romero et al., 2014). For the response-based method, the student model is trained from the teacher's output. In the feature-based method, students are trained from the teacher's intermediate output and/or weights. Distillation methods can be categorized into online (Zhang et al., 2018) and offline distillation (Ba and Caruana, 2014; Hinton et al., 2015), depending on whether the teacher parameters are updated while the students are learning. In our study, we cannot access the teacher model. Therefore we transfer response-based knowledge to a student through offline distillation. Response-based knowledge refers more explicitly to information propagated through the teacher's output probability. Suppose the output probabilities are not included in the submission results of the shared task, as in this paper. In that case, the teacher's knowledge can also be extracted from their predictions for additional unlabeled data.

2.2 Semi-Supervised Learning

The learning method used in our scheme can be classified as a semisupervised method such as Self-Training (Yarowsky, 1995) and Co-Training (Blum and Mitchell, 1998) in that it uses unlabeled data predictions for learning. Self-Training is a method for building a more robust machine learning model by adding labels with high confidence to the training data from trained model predictions and retraining the model. Co-Training is an extension of the Self-Training method, where the instances added to the training data are determined by the label confidence obtained using two or more models. Those methods are an approach that combines a small and large amount of labeled and unlabeled data, respectively, during model training. However, to the best of our knowledge, no study has used the results of a shared task to extend the training data. We try simple self-training in Sec. 4.3 to show the benefits of extending the training data with the results of a

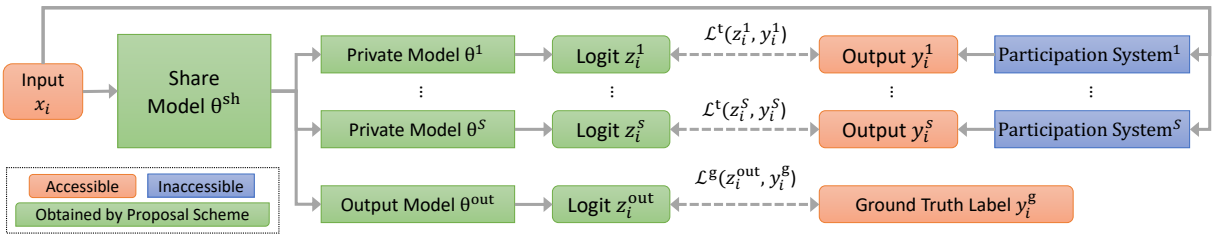

Figure 1: Overview of the Teacher-Student architecture used in Co-Teaching. Here, we do not have access to the systems that participated in the shared task, but we have access to their submission results. In order to reproduce the inaccessible participation systems, we treat those results as teachers and train a student model.

shared task on unlabeled data.

3 Proposal Scheme

We propose a "Co-Teaching" scheme to address the system submission, which is problematic due to license protection and operating environment issues in the shared task, even when the system developed by the task participant is valuable. In the proposed scheme, the systems submitted to the shared task are considered teachers, and a new student model is trained through their outputs. We expect that this scheme allows us to build a system that comprehensively and integrally reproduces the teacher characteristics. The training data distributed to participants in the shared task can also be used for student learning. This scheme can be performed even after completing the shared task, as long as the training data distributed in the task and the submission results are available. However, it is better to have more information about the teachers available for training the students. It is desirable to be able to use the prediction probabilities assigned to the teacher outputs, as well as training and unlabeled data predictions. Therefore, cooperation from shared tasks is essential for this scheme to work effectively.

An overview of the teacher-student architecture used in the scheme is presented in Fig. 1. Let us assume that we have the outputs $\{\mathcal{Y}_s\}_{s\in[S]}$ of the participation system for the input space $\mathcal{X}$, where S is the number of participation systems. Each data point can be described as $\{x_i, y_i^1, ..., y_i^S\}_{i\in[N]}$, where N is the number of data instances, x_i is i-th instance, and y_i^s is the output of s-th teacher for i-th instance. Some instances also have $\{y_i^g\}_{i\in[M],M\leq N} \in \mathcal{Y}$ ground truth labels that were used to train the teachers. In this scheme, our goal is to learn the student model $\{\theta^{sh}, \theta^1, ..., \theta^S, \theta^{out}\}$, where θ^{sh} is the shared parameter to reproduce the features common among the teachers, θ^s is the private parameter to reproduce each teacher output, and θ^{out} is the parameter to output the overall prediction results. Here, we simultaneously reproduce each teacher model $f(x;\theta^{sh},\theta^s) : \mathcal{X} \mapsto \mathcal{Y}$ and learn the overall output from the labeled data $f(x;\theta^{sh},\theta^{out}) : \mathcal{X} \mapsto \mathcal{Y}$. The loss function used for learning is written as

$$\hat{\mathcal{L}}(\theta^{sh}, \theta^1, ..., \theta^S, \theta^{out}) = \begin{cases} \frac{1}{2}(\alpha\hat{\mathcal{L}}^t(\theta^{sh}, \theta^1, ..., \theta^S) + \hat{\mathcal{L}}^g(\theta^{sh}, \theta^{out})) & (i \leq M) \\ \hat{\mathcal{L}}^t(\theta^{sh}, \theta^{out}) & (\text{otherwise}), \end{cases}$$

where $\hat{\mathcal{L}}^t(\theta^{sh}, \theta^1, ..., \theta^S) = \frac{1}{S}\sum_{s=1}^{S}\hat{\mathcal{L}}_s^t(\theta^{sh}, \theta^s)$,

$$\hat{\mathcal{L}}_s^t(\theta^{sh}, \theta^s) = \mathcal{L}^t(z_i^s, y_i^s),$$

$$\hat{\mathcal{L}}^g(\theta^{sh}, \theta^{out}) = \mathcal{L}^g(z_i^{out}, y_i^g),$$

$$z_i^s = f(x_i;\theta^{sh},\theta^s),$$

$$z_i^{out} = f(x_i;\theta^{sh},\theta^{out}).$$

α is a weight that determines the balance of loss between the ground truth label and the teacher output, respectively. We train the student model by minimizing the above loss.

Since we also utilize the logit z_i^s of each private layer other than the logit z_i^{out} of the output layer, the final prediction probability p_i of the entire model is calculated as follows:

$$p_i = \frac{1}{2}(p_i^{out} + p_i^{mean}),$$

where $p_i^{out} = \text{softmax}(z_i^{out})$,

$$p_i^{mean} = \frac{1}{S}\sum_{s=1}^{S}\text{softmax}(z_i^s).$$

...現在の ...The current	佐鳴湖公園 Sanaruko Park *Park*	は以前よりもかなり狭くなっている... is much smaller than before....
...水深は護岸下で ...The water depth is	およそ2メートル approximately two meters *Water depth*	である ... under the revetment...
...周囲は浜松市の ...The surrounding area is known as a	サクラ cherry blossom *Plants*	の名所として知られ、... spot in Hamamatsu City, ...

Figure 2: Examples of attribute values of *Lake* category.

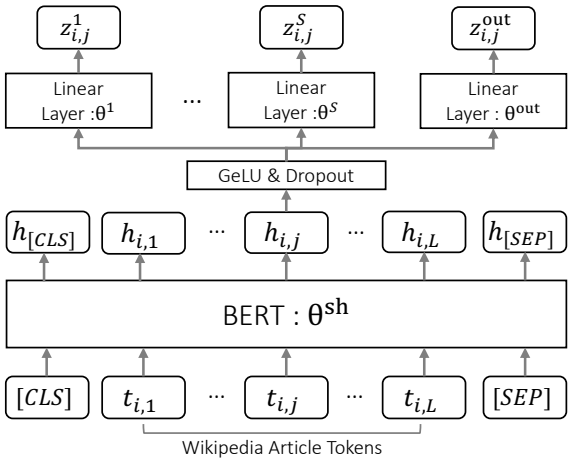

Figure 3: Structure of the student model used in the experiment.

4 Experiment

4.1 SHINRA2019-JP

SHINRA2019-JP is a shared task to extract the attribute values in Japanese Wikipedia articles. These articles are preclassified into Extended Named Entity (ENE) categories by Suzuki et al. (2018). ENE is a set of Named Entity types defined by Satoshi (2008) and includes about 200 categories. The attributes are predefined for each category by ENE, and the participants build attribute value extraction systems using the distributed training portion of Wikipedia and are instructed to make their predictions for all remaining Wikipedia articles. The task requires specifying where the mention of the value occurs and not just extracting the surface text. The SHINRA2019-JP targets 35 categories; five categories called *JP-5* subclass are those previously targeted in SHINRA2018-JP in addition to 30 new categories. Of the 30 categories, 14 belong to the *Location* subclass, and the rest belong to the *Organization* subclass. Examples of attributes and values in the *Lake* category of the *Location* subclass are shown in Fig. 2. Note that in this study, we used only 33 out of the 35 categories in SHINRA2019-JP because the two categories have no test data.

In this task, participants can access the manually annotated training data and articles in each category. For the SHINRA shared task evaluation, a portion of the data is hidden, and all the participants have to annotate all the data so that the organizer can create unified data for all Wikipedia entries.

A total of nine teams participated in the SHINRA2019-JP task. Some participants submit results for a subset of the categories, and six to nine systems submit results for every category. Various methods are used, including rule-based methods, the ML method using CRF and SVM, a deep learning-based method, and DrQA (Chen et al., 2017).

This task follows the Resource by Collaborative Contribution (RbCC) scheme. Therefore, the task organizers release all submission results as a resource. In this task, participants do not necessarily need to submit the prediction probabilities assigned to the system outputs; thus, the organizers do not share these values. The organizer also distributed the development data for the City and Lake categories, which are not used to evaluate. In our study, we use those labels for our detailed analysis. The task organizers have not released the evaluation set used in the task. Therefore, we sent our results to the organizers and received the evaluation results.

4.2 Co-Teaching on Shared Task

In order to demonstrate the effectiveness of the proposed scheme, we use the submission results shared by SHINRA2019-JP to train a student model. Although this is an attribute value extraction task, it can be solved as a sequence labeling task because each attribute value contains the offset of its occurrence in the text. We use the IOB2 (Tjong Kim Sang and Veenstra, 1999) scheme to solve the sequence labeling task. That is, we classify

the first word of an attribute value as Beginning (B), the following words as Inside (I), and words outside the attribute value as Outside (O). In this task, a word may have multiple attribute labels. Therefore we use I, O, and B tags for each attribute. More specifically, we classify the word $\{t_{i,j} \in x_i\}_{j \in [K]}$ into $y^{\mathrm{g}}_{i,j}$ and $y^{s}_{i,j,\forall s \in [S]}$, where $|y^{\mathrm{g}}_{i,j}| = C$, $|y^{s}_{i,j,\forall s \in [S]}| = C$, K is the sentence length, and C is the number of attributes to be extracted. We use MeCab[2] to tokenize Japanese text.

Subclass	Category	Num. Pages	Num. Train	Num. Attributes
JP-5	Airport	1,615	599	24
	City	51,035	1,000	25
	Company	35,356	995	34
	Compound	5,819	598	15
	Person	308,610	999	21
Location	Bay	354	200	28
	Continental Region	269	147	15
	Country	1,304	158	22
	Domestic Region	2,054	200	13
	Geological Region Other	2,269	200	19
	GPE Other	395	200	18
	Island	2,292	173	34
	Lake	772	200	32
	Location Other	2,525	200	18
	Mountain	3,718	200	32
	Province	12,008	198	26
	River	2,764	200	17
	Sea	291	200	28
	Spa	1,080	190	21
Organization	Company Group	386	200	28
	Ethnic Group Other	1,133	200	13
	Family	1,904	200	18
	Government	3,053	200	20
	International Organization	949	191	20
	Military	3,368	200	22
	Nonprofit Organization	5,046	200	23
	Organization Other	3,867	183	13
	Political Organization Other	1,177	200	12
	Political Party	1,543	199	23
	Show Organization	10,290	196	22
	Sports Federation	790	200	23
	Sports League	841	189	24
	Sports Team	4,828	199	29

Table 1: Distribution of SHINRA2019-JP data.

Furthermore, we define the student model used in this experiment, as shown in Fig. 3. We use BERT-base (Devlin et al., 2019) for θ^{sh} and a linear layer for $\theta^1, ..., \theta^S, \theta^{\mathrm{out}}$, respectively, and apply Dropout (Srivastava et al., 2014) and an activation function GeLU (Hendrycks and Gimpel, 2020) to the output of BERT. BERT is pretrained using the same scheme as RoBERTa (Liu et al., 2019) utilizing Japanese Wikipedia. Class Balanced Focal Loss (Cui et al., 2019), which is a combination of Class Balanced Loss (Cui et al., 2019) and Focal Loss (Lin et al., 2017), is used for the loss functions $\hat{L}^{\mathrm{t}}$ and $\hat{L}^{\mathrm{g}}$ to deal with the class imbalance IOB2 labels. We also determine α, which balances the loss between the ground truth label and the teacher outputs, as $\alpha = \frac{\sum_{i=1}^{N} |x_i|}{\sum_{i=1}^{M} |x_i|}$. This weight value equalizes the impact of the two losses on the entire dataset.

In this task, 269 to 308610 articles are available for each category, and 147 to 1000 have ground truth labels. All articles have the system results that participated in each category. For reducing the computational cost, we limited the number of articles to 2000, including all labeled data. The detailed data statistics are shown in Table 1. A student model is trained for each category, and we use 10% of the labeled data as development data and the rest, including all unlabeled data, as training data. Models are trained for each category.

In this experiment, we also fine-tune BERT using only the training data of SHINRA2019-JP. This is called the Non-Teaching setting. By comparing Co-Teaching and Non-Teaching, we can separate the advantages of the proposed scheme and the model structure. Moreover, we integrate the predictions for unlabeled data of the model obtained in the Non-Teaching setting with the training data and retrain the model. This setting is similar to the self-training setting, so we temporarily call it Self-Teaching. By comparing Co-Teaching and Self-Teaching, we can separate the advantage of the proposed method from input expansion using unlabeled data. That means we can only evaluate the advantage gained by the knowledge extracted from the system outputs that participated in the shared task. These comparisons are a kind of ablation test.

We use the Adam optimizer to train the model in each setting and use mixed precision for computational efficiency. We also determine the batch size and the γ used for Class Balanced Loss by grid search from $\{8, 16, 32\}$ and $\{0.999, 0.9999, 0.99999\}$, respectively, and obtain the class statistics used for Class Balanced Loss from the ground truth labels. The other parameters used in this experiment are: Adam learning rate $\alpha_{lr} = 5 \times 10^{-5}$, Adam $\beta_1 = 0.9$, Adam $\beta_2 = 0.999$, and Adam $\epsilon = 10^{-8}$. When we use 2000 articles for training, once training takes about 10 hours on a GPU when the batch size is eight and about five hours on eight GPUs, the batch size is 32. Here, all GPUs are NVIDIA Tesla V100.

4.3 Experimental Results

The experimental results are listed in Table 2. Each score is the across-category macro-average F1 value for each subclass, and the score for each cate-

[2]https://taku910.github.io/mecab/

Subclass	Team ID					Non-Teaching	Self-Teaching	Co-Teaching
	02	03	05	07	10			
JP-5	67.99	-	57.60	63.93	68.40	68.22	68.76	**69.95**
diff	*-1.96*	-	*-12.35*	*-6.02*	*-1.55*	*-1.73*	*-1.19*	-
Location	59.51	57.89	49.56	53.68	58.21	57.74	58.43	**63.63**
diff	*-4.12*	*-5.75*	*-14.08*	*-9.95*	*-5.42*	*-5.89*	*-5.21*	-
Organization	51.33	53.68	37.52	48.70	**57.91**	51.14	52.14	57.55
diff	*-6.22*	*-3.88*	*-20.03*	*-8.85*	*+0.36*	*-6.41*	*-5.41*	-
All	57.33	-	45.67	53.12	59.63	56.53	57.32	**62.01**
diff	*-4.69*	-	*-16.34*	*-8.89*	*-2.38*	*-5.48*	*-4.69*	-

Table 2: Experimental results for each subclass on SHINRA2019-JP. *diff* means the difference between the proposal scheme. **Bold number** indicates the highest score, and underlined number specifies the highest score within the participation systems.

gory is the across-attribute micro-average F1 value. For equal comparison, system results that did not submit predictions for all categories belonging to a subclass were temporarily excluded from the table. We can see from the table that Co-Teaching obtained a better score than the best system in the JP-5 subclass and Location subclass. In particular, we can see a great improvement in the Location subclass (i.e., 4.12 points) compared to the best system. The significant difference in scores between the Non-Teaching and Co-Teaching results signifies that this improvement was not obtained due to the model structure's advantages. In addition, Self-Teaching is slightly superior to Non-Teaching, which may be due to the effect of input expansion using unlabeled data. However, Self-Teaching is also significantly inferior to Co-Teaching. This difference suggests that the knowledge derived from the participating systems is more valuable than the input extension. When we focus on the Non-Teaching results in the Organization subclass, the difference in scores between the best system is -6.77, which is significantly inferior. This difference suggests that either the BERT model's structure is incompatible with the Organization subclass or the participants may have used additional knowledge about the Organization subclass. However, in Co-Teaching, the score is equivalent to the best system in the Organization subclass by learning from the participating system results. The overall Co-Teaching score is better than the best system (Team ID:10) score. This best system consists of two BERT that take plain text input as used in this study or HTML text input recovered from the Wikipedia dump. Therefore, the student model performs better than the teacher model, even though less information is given to each input instance. This result implies that the system can be obtained indirectly through the proposed scheme without requiring the shared task participants to submit their systems, even if they use additional knowledge. In addition, the score improvement is based on the knowledge gained from the systems other than the best one, demonstrating the potential usefulness of those systems. This result motivates task participants.

The scores for each category are displayed in Table 3. The method of calculating the score is the same as in Table 2. *Best System* means the best score from the system results. In this task, five teams seem to have achieved the best score in one or more categories. In order to obtain the overall best system for this task, we need to require all five teams to submit their systems. However, the Co-Teaching score is equivalent to the average of the best systems. This result shows the effectiveness of the proposed scheme.

In all categories, Co-Teaching performed better than Non-Teaching. Teachers do not negatively affect student learning in this situation, as Best System is better than Non-Teaching in most categories.

The table confirms that Co-Teaching performed worse than Best System in 15 out of 33 categories. The correlation between the best and second-best system difference and the improvement from the Best System with Co-Teaching is shown in Fig. 4. The correlation coefficient between these two values is $r = -0.519$, indicating a negative correlation.[3] In situations where only a single teacher is superior, the student model is not learning well. In this study, the losses are averaged from the teachers we use for student learning, so the loss of a single

[3]This result ($p = 1.99 \times 10^{-3}$) is statistically significant at $p < 0.01$ with t-test.

Subclass	Category	Team ID									Best	Non-	Self-	Co-
		01	02	03	04	05	06	07	08	10	System	Teaching	Teaching	Teaching
JP-5	Airport	44.15	89.55	79.92		86.03		84.74		88.03	89.55	86.69	86.69	**90.49**
	City	7.96	66.40	60.93		56.19		62.37		66.49	**66.49**	64.16	65.75	65.88
	Company	11.92	61.95	63.25		39.59	10.48	53.82		66.13	**66.13**	57.62	58.14	58.49
	Compound		45.67	47.98		50.32		49.35	50.72	49.04	50.72	56.53	57.26	**58.69**
	Person	3.42	76.40		34.53	55.88		69.39		72.31	**76.40**	76.09	75.95	76.19
Location	Bay	0.16	67.47	60.86		52.58		58.55		58.73	67.47	63.60	63.89	**67.62**
	Continental Region		56.38	53.71		41.48		51.03		53.28	56.38	52.99	53.36	**59.87**
	Country		57.66	61.27		48.87		52.00		64.26	64.26	57.82	60.43	**65.70**
	Domestic Region		48.55	43.09		23.91		44.71		50.49	50.49	42.21	44.42	**52.38**
	Geological Region Other	2.91	57.29	58.98		43.83		46.77		62.65	**62.65**	52.01	53.27	60.45
	GPE Other	0.77	56.45	48.71		38.03		46.38		49.07	**56.45**	52.07	52.60	55.85
	Island		67.40	66.31		53.87		58.90		59.77	67.40	63.34	63.46	**70.29**
	Lake	9.67	63.09	59.63		55.42		57.01		43.43	63.09	60.50	60.50	**67.47**
	Location Other	2.40	40.70	40.82		38.35		41.10		49.00	49.00	43.84	43.85	**50.37**
	Mountain	4.18	62.73	59.24		56.62		56.27		61.46	62.73	58.30	58.44	**64.59**
	Province	2.10	66.18	60.45		60.42		59.39		67.25	67.25	67.02	68.25	**71.14**
	River	3.52	59.49	61.30		48.81		56.24		64.92	**64.92**	61.67	61.84	64.73
	Sea		60.96	62.90		57.35		55.42		65.65	**65.65**	61.82	62.25	64.15
	Spa	10.82	68.83	73.18		74.24		67.75		65.00	74.24	71.18	71.42	**76.28**
Organization	Company Group	0.57	56.42	65.03		29.72		54.32		61.55	65.03	58.32	59.47	**65.49**
	Ethnic Group Other		51.05	50.56		39.96		47.99		56.71	56.71	49.85	50.97	**57.54**
	Family	0.18	62.78	60.40		39.09		61.86		69.66	**69.66**	58.79	60.32	67.97
	Government	2.75	50.24	51.20		43.07		47.63		47.09	51.20	49.61	50.66	**56.95**
	International Organization	2.31	48.58	52.71		39.62		44.08		51.97	52.71	49.43	49.31	**57.45**
	Military	1.94	53.14	60.12		39.55		52.67		67.52	**67.52**	56.43	59.03	63.97
	Nonprofit Organization	3.23	46.96	47.53		39.88		40.20		59.75	**59.75**	44.50	46.39	53.99
	Organization Other	4.06	50.46	53.95		42.22		42.74		52.87	**53.95**	48.34	47.04	51.05
	Political Organization Other		40.60	34.70		21.35		26.64		47.55	**47.55**	35.58	35.34	42.82
	Political Party	1.35	46.65	47.48		39.78		41.49		52.24	**52.24**	44.05	44.72	49.79
	Show Organization	1.17	63.96	71.43		36.32		64.80		68.33	**71.43**	64.98	65.22	69.26
	Sports Federation	4.52	51.39	56.94		46.21		50.90		56.81	56.94	51.15	51.84	**57.47**
	Sports League	2.03	45.65	47.97		24.88		58.42		63.86	**63.86**	52.91	55.32	53.95
	Sports Team	3.91	50.68	51.44		43.69		48.09		54.92	54.92	52.05	54.28	**58.04**
	Macro Average	-	57.33	-	-	45.67	-	53.12	-	59.63	61.96	56.53	57.32	**62.01**

Table 3: Experimental results for each category on SHINRA2019-JP. **Bold number** represents the highest score in the right four columns, and underlined number designates the highest score in the participation systems.

superior system may be dominated by other systems. For further improvement, we may need to apply methods such as MGDA (Sener and Koltun, 2018) used in multitask learning to balance teacher losses during student learning dynamically.

Our study limited the data used for learning the student model to 2000 articles in each category due to the computational cost. However, there are much more articles available for some categories. Therefore, we studied the score variance of the student model in the City category as we used more articles for training. The score for each output of the student model is also tracked. We use the data for analysis in the City category for this experiment. As in other experiments, the batch size and γ used for Class Balanced Loss is determined using a grid search. The batch size is determined from $\{8, 16, 32\}$ when the number of articles used for training is 2000, $\{20, 40, 80\}$ when the number of articles is 5000, and $\{40, 80, 160\}$ when the number of articles is 10000. Also, γ was determined from $\{0.999, 0.9999, 0.99999\}$.

The results are shown in Fig. 5. The model's final output result is better when 5000 articles are used for training compared to 2000 articles. However, when 10000 articles are used for training, the score drops. In this study, we have determined α to make the effect of these two losses equivalent across the dataset. Therefore, as the unlabeled data increases, the loss between the teacher and student for each instance becomes much smaller than the loss between the ground truth label and the student. When using large amounts of unlabeled data, it may be necessary to constrain α. Surprisingly, the average score of private layers is consistently higher than the output layer trained on ground truth. This result signifies that the information obtained from the teachers is more valuable than the ground truth label. However, using the appropriate α, both outputs complement the final output, as in the case of using 5000 articles. In future work, we examine how to find the appropriate α corresponding to the proportion of unlabeled data.

5 Discussion

In order to apply the Co-Teaching scheme to a shared task, it is required to disclose the participants' results. The task organizer needs to obtain permission to publish the results in advance, which may become an obstacle for the participants. How-

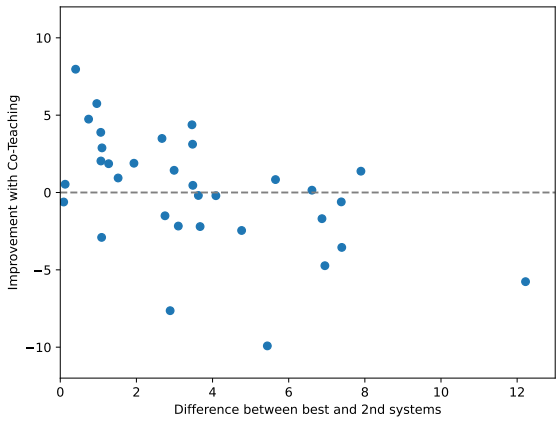

Figure 4: Correlation between the best and second-best system difference and the improvement from the best system with Co-Teaching.

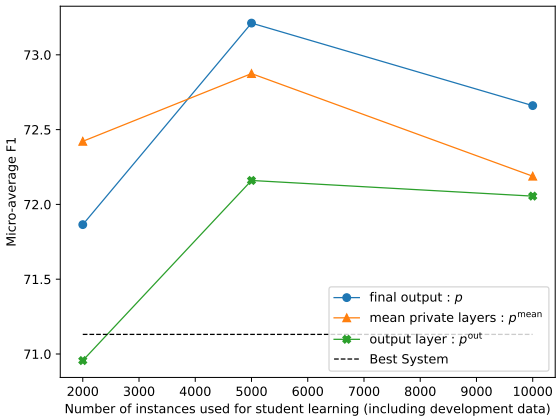

Figure 5: Changes in student model scores on data for analysis in City category when the number of articles used to train the student model is increased.

ever, the disclosure of the results should be a much lower burden than the system disclosure. For example, a dictionary of a Named Entity Recognition task may be a valuable resource for the participant's organization. Therefore, it could be hard for the organization to disclose the system with the dictionary. However, the task's outputs may be much less valuable for the participant's organization, as it is challenging to reproduce the dictionary from the system results.

Suppose a Co-Teaching scheme is to be incorporated into a shared task. In that case, it must be possible to design the student model technically, and it is also desirable that a large amount of unsupervised data is easily obtainable. We discuss the shared tasks that have been implemented in the past from those perspectives. We believe that the conditions are satisfied for the classification tasks such as the Sentiment Analysis and Relation Classification tasks in SemEval (Hendrickx et al., 2010), Dialect Classification task in NADI (Abdul-Mageed et al., 2021), and word classification task in CoNLL (Tjong Kim Sang and De Meulder, 2003). Also, the conditions are satisfied for translation tasks such as those in WMT (Barrault et al., 2020) and WAT (Nakazawa et al., 2020) as well as generation tasks such as those in SDP (Chandrasekaran et al., 2020) and FNS (El-Haj et al., 2020). The similarity between these tasks is that the formats of the training data and the task submissions are essentially the same. That is, a student model can be designed with few modifications to the model designed for the training data (e.g., by adding an output layer or decoder for each participating system). However, there are cases where the format of the training data and the task submission are different. For example, the training data in the IWSLT speech translation task (Ansari et al., 2020) consists of the end-to-end speech translation or the transcription dataset and the bilingual corpus, but only the final target-language text is submitted. In this case, if the participant uses the latter non-end-to-end dataset, the dataset used and the task submission format are different. However, the task organizers can solve this formatting problem by requesting the participants to submit the transcribed text. In the above tasks, except for the relational classification task requiring entity pair information, the unlabeled data, such as plain text or speech, is easily obtainable. The prediction results of the systems developed by task participants for those unlabeled data help make the Co-Teaching scheme work effectively, as shown in the experiments in this paper. As mentioned above, many shared tasks are suitable for the application of the proposed scheme.

We have discussed the design of the student model and the data required in the Co-Teaching scheme, but whether the student model can successfully reproduce the submission system needs to be shown in many future experiments. For these experiments, we need the submission results of many shared tasks. In addition, although it was not available in this experiment, if the output probability of the system is available, further improvement can be expected using the KL-divergence and other methods. We hope that these data become more open in the future.

We have given approximate computational costs at the end of Sec. 4.2. As shown, the effort and

cost involved in implementing this scheme are not trivial because we need to build a student model with sufficient capacities to apply the Co-Teaching scheme, such as BERT. However, we believe that these costs and efforts are much smaller than the total costs and efforts spent by the shared task participants.

The Teacher-Student architecture that we used in this study is simple. Nevertheless, we were able to demonstrate the usefulness of the Co-Teaching scheme. However, there is room for improvement in the architecture, e.g., preventing score degradation in cases where only a single system is superior. In the future, we would like to compare knowledge distillation methods and develop a more suitable architecture for the Co-Teaching scheme. Also, we aim to conduct detailed validation of our proposed scheme by ablation tests with the removal of each system and stress tests with the addition of noise systems.

Finally, we would like to introduce research with similar motivations. Potthast et al. (2019) developed an architecture for shared tasks called TIRA, consisting of virtual environments for system development and evaluation modules. On TIRA, each participant of the shared task develops a system on the given virtual environment and receives an evaluation by submitting the system to the evaluation module. In this process, TIRA stores the systems that have been submitted. But, third parties cannot access such systems directly. Instead, TIRA provides API, and they can receive the system results for any input via API. Thus, the systems are public virtually, but licensing issues are minimized as long as the whole system or part of it does not leak out of TIRA. Although this research has a different focus, integrating the Co-Teaching scheme into the TIRA architecture would allow seamless student model learning and efficient leveraging of the participants' efforts.

6 Conclusion

In this paper, we proposed a new scheme for shared tasks called "Co-Teaching." It is a scheme to build a single system from the participants' outputs under the Teacher-Student architecture. We conducted an experiment based on the SHINRA2019-JP shared task to demonstrate the effectiveness of our scheme. As a result, we were able to construct a system that was 2.38 points higher in F1-value than the best participating system. We hope that this scheme will be applied to many shared tasks to utilize the participant's efforts effectively. Furthermore, we believe the shared tasks can be a more useful scheme if it is not only the place for the optimization of the given task but the outcome is designed so that some resource is obtained, such as a Knowledge Base (RbCC) or a superior system can be created from the participant's system collection (Co-Teaching).

Acknowledgements

This work was supported by JST, ACT-X Grant Number JPMJAX20AI, Japan and JSPS KAKENHI Grant Number JP20269633 and JST, PRESTO JPMJPR20C.

References

Muhammad Abdul-Mageed, Chiyu Zhang, Abdel-Rahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondřej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, Alexander Waibel, and Changhan Wang. 2020. FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 1–34, Online. Association for Computational Linguistics.

Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, pages

92–100, New York, NY, USA. Association for Computing Machinery.

Muthu Kumar Chandrasekaran, Guy Feigenblat, Eduard Hovy, Abhilasha Ravichander, Michal Shmueli-Scheuer, and Anita de Waard. 2020. Overview and insights from the shared tasks at scholarly document processing 2020: CL-SciSumm, LaySumm and LongSumm. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 214–224, Online. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Association for Computational Linguistics (ACL)*.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples.

Hoa Dang and Karolina Owczarzak. 2009. Overview of the tac 2008 update summarization task.

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.

Mahmoud El-Haj, Ahmed AbuRa'ed, Marina Litvak, Nikiforos Pittaras, and George Giannakopoulos. 2020. The financial narrative summarisation shared task (FNS 2020). In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 1–12, Barcelona, Spain (Online). COLING.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Dan Hendrycks and Kevin Gimpel. 2020. Gaussian error linear units (gelus).

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

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.

Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Shohei Higashiyama, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, and Sadao Kurohashi. 2020. Overview of the 7th workshop on Asian translation. In *Proceedings of the 7th Workshop on Asian Translation*, pages 1–44, Suzhou, China. Association for Computational Linguistics.

Iadh Ounis, Craig Macdonald, and Ian Soboroff. 2008. Overview of the TREC-2008 blog track. page 14.

Martin Potthast, Tim Gollub, Matti Wiegmann, and Benno Stein. 2019. *TIRA Integrated Research Architecture*, pages 123–160. Springer International Publishing, Cham.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Y. Bengio. 2014. Fitnets: Hints for thin deep nets. *arXiv*.

Sekine Satoshi. 2008. Extended named entity ontology with attribute information. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Satoshi Sekine, Akio Kobayashi, and Kouta Nakayama. 2019. SHINRA: Structuring wikipedia by collaborative contribution. In *Submitted to Automated Knowledge Base Construction*.

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Beth M. Sundheim. 1995. Overview of results of the MUC-6 evaluation. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Masatoshi Suzuki, Koji Matsuda, Satoshi Sekine, Naoaki Okazaki, and Kentaro Inui. 2018. A joint neural model for fine-grained named entity classification of wikipedia articles. *IEICE Transactions on Information and Systems*, E101.D(1):73–81.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th Conference*

on Computational Natural Language Learning - Volume 7, ConLL '00, pages 127–132, USA. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 173–179, Bergen, Norway. Association for Computational Linguistics.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. 2018. Deep mutual learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4320–4328.