Revisiting the Evaluation of End-to-end Event Extraction

Shun Zheng[†], **Wei Cao**[†], **Wei Xu**[‡], **Jiang Bian**[†]
[†]Microsoft Research
[‡]Institute of Interdisciplinary Information Sciences, Tsinghua University
{shun.zheng, wei.cao, jiang.bian}@microsoft.com;
weixu@mail.tsinghua.edu.cn

Abstract

Event extraction (EE) aims to harvest event instances from plain text, where each instance is composed of a group of event arguments with specific event roles. Existing end-to-end EE research usually adopts the role-averaged evaluation that produces evaluation measures by averaging evaluation statistics of each event role. However, although this averaged metric can indicate the model performance to some extent, we find that such metric can be pretty misleading to downstream applications that utilize an event instance as a whole, where one wrongly identified event argument can substantially alter the whole meaning of an event instance. To mitigate this gap and provide a more complete understanding of performance, we propose two new evaluation metrics that also consider an event instance as a whole and explicitly penalize wrongly identified event arguments. Moreover, to support diverse preferences of evaluation metrics motivated by different scenarios, we propose a new training paradigm based on reinforcement learning for a typical end-to-end EE model, i.e., Doc2EDAG. Our extensive experiments show that the new training improves the initial one by a large margin (about 10%) under new metrics. Nevertheless, the current performance is still far from satisfactory, and optimizing towards these new metrics calls for more future research.

1 Introduction

Event extraction (EE) is a vital task that aims to harvest structured event instances from unstructured plain text. Such structured knowledge can benefit many downstream applications, such as question answering, language understanding, knowledge graph, etc. In general, an *event instance* is composed of a group of entities (person, organization, date, etc.) that jointly describes an incident. Each entity of the event instance, also referred to as the *event argument*, plays a specific *event role*. Multiple event instances of the same event type populate an *event table*.

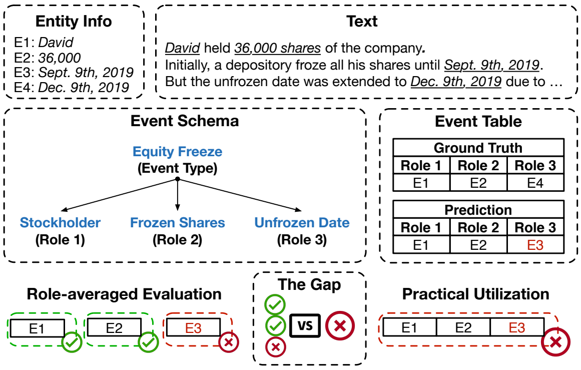


Figure 1: An example to illustrate the gap between the *role-averaged* evaluation and a practical application that utilizes an event instance as a whole, where "E1", "E2", "E3", and "E4" are entity marks, "Role 1", "Role 2", and "Role 3" represent event roles, and "(E1, E2, E4)" is an event instance. We can see that the wrong argument "E3" makes the whole event instance "(E1, E2, E3)" unrealistic, but the *role-averaged* evaluation still regards "E1" and "E2" as correct ones.

The early method (Ahn, 2006) formalized EE as the unification of many sub-tasks, including entity recognition, event detection, and argument extraction, etc. Later research improved EE from two aspects: the modeling to capture complicated semantic structures (Li et al., 2013; Yang and Mitchell, 2016; Nguyen and Nguyen, 2019) and the labeling to combat the lack of training data (Chen et al., 2017; Zeng et al., 2018). Recently, Zheng et al. (2019a) proposed the first end-to-end model, called Doc2EDAG, for document-level EE. Given a text document, Doc2EDAG can generate an entity-based directed acyclic graph (EDAG) to fill an event table directly.

Different from the modeling and labeling aspects, the evaluation of EE attracted very little re-

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4609–4617
August 1–6, 2021. ©2021 Association for Computational Linguistics

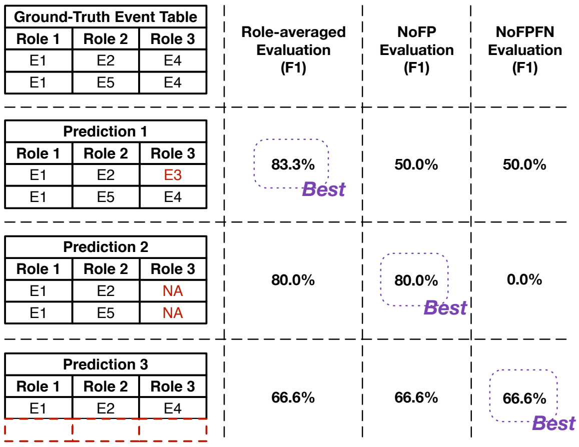
search attention. Notably, most existing research, merely presenting a separate evaluation for each sub-task of EE, can only get the approximation of the overall performance by combining those fragmentary evaluation results. Even the latest work (Zheng et al., 2019a), reporting the overall performance of EE, still followed the traditional approximate measure by averaging evaluation statistics of each event role. We refer to this kind of approximation as the *role-averaged* evaluation. However, many downstream applications need to utilize an event instance as a whole, where a wrongly identified argument can substantially change the meaning of an event and cause severe misleading effects. Figure 1 presents an example from the financial domain, where the downstream application cannot utilize an event instance with an incorrect date argument for decision making, but the *role-averaged* evaluation still assigns this example with two true-positive arguments. Moreover, we note that this *event-as-a-whole* demand is a common case that widely exists in many other domains, such as legislation, health, etc.

To enable the evaluation support for the *event-as-a-whole* scenario and provide a more complete understanding of performance, we propose two new metrics that directly make judgments on an event instance rather than averaging the performance of its arguments. The first metric is *NoFP*, which regards a predicted event with any false-positive (FP) error at the entity level as an event-level FP error. The second one is *NoFPFN*, which permits neither FP errors nor false-negative (FN) errors at the entity level for a predicted event being considered as a true-positive (TP) one. In practice, we can choose to use the proper metric according to the specific scenario. For example, if we only care about the correctness of predicted events, we can utilize the *NoFP* evaluation that penalizes FP entities explicitly. If we further pursue the completeness, we can utilize the *NoFPFN* evaluation.

The necessity of employing new evaluation metrics, however, raises a dilemma in training effective EE models. On one side, since the *role-averaged* metric is inconsistent with *NoFP* or *NoFPFN*, training towards the *role-averaged*, as did by traditional methods, may not lead to improvements under new metrics. For instance, Figure 2 illustrates such inconsistence between different evaluation settings. We can observe that 1) the first prediction is the best one under the *role-averaged* evaluation but is sub-optimal for other settings, 2) the second prediction best fits the *NoFP* evaluation but suffers the most errors under the *NoFPFN* evaluation, and 3) the third prediction makes the most mistakes at the entity level but achieves the best performance under the hardest *NoFPFN* evaluation. In fact, existing methods for EE, including the most recent Doc2EDAG, merely optimized model parameters towards proper predictions for each event role separately. Thus, they failed to align with new metrics.

On the other side, optimizing these new metrics is non-trivial. First, it is hard to design appropriate training objectives since those metrics are non-differentiable. Moreover, the supervision for role-level predictions is inevitably delayed because we cannot calculate event-level metrics until obtaining predictions for all event roles. To address these challenges, we propose a new reinforcement learning (RL) paradigm of Doc2EDAG by regarding EDAG generation as a Markov decision process. In this way, we can optimize model parameters with the guidance of a delayed reward, which can be specified by a specific metric. At the same time, we realize that this shift of training also bridges the gap between training and inference because RL forces the model to generate EDAGs during training. Extensive experiments demonstrate that our RL-based training improves the vanilla one significantly (about 10%) under new metrics.

We summarize our contributions as follows.



Ground-Truth Event Table

Role 1	Role 2	Role 3
E1	E2	E4
E1	E5	E4

Prediction 1

Role 1	Role 2	Role 3
E1	E2	E3
E1	E5	E4

Prediction 2

Role 1	Role 2	Role 3
E1	E2	NA
E1	E5	NA

Prediction 3

Role 1	Role 2	Role 3
E1	E2	E4

	Role-averaged Evaluation (F1)	NoFP Evaluation (F1)	NoFPFN Evaluation (F1)
Prediction 1	83.3% Best	50.0%	50.0%
Prediction 2	80.0%	80.0% Best	0.0%
Prediction 3	66.6%	66.6%	66.6% Best

Figure 2: We present a ground-truth event table with two event instances (“(E1, E2, E4)” and “(E1, E5, E4)”) and three different predictions to show diverse preferences of those evaluation settings, where all marks follow the meanings in Figure 1, “NA” denotes an empty argument, we color wrong or missed event arguments as red, and we mark the best prediction for each evaluation setting.

- We revisit the evaluation of EE to support downstream applications that need to utilize an event instance as a whole. Specifically, we propose two new metrics to provide a more complete understanding of performance.

- We propose a new training paradigm for a typical end-to-end EE model, Doc2EDAG, to enable the flexible adaptation to new metrics with hard constraints.

- Our empirical studies show that under the *event-as-a-whole* scenarios, the traditional *role-averaged* evaluation tends to severely overestimate the performance. Moreover, the initial training scheme performs poorly on new metrics, while our RL-based training improves it significantly (about 10%).

2 Related Work

Since EE is a very sophisticated task that requires the unification of many sub-tasks (Ahn, 2006), including entity recognition, event detection, and argument extraction, plenty of previous research put considerable efforts to the modeling aspect.

Specifically, Nguyen and Grishman (2015); Liu et al. (2017); Chen et al. (2018); Wang et al. (2019); Liu et al. (2019) only considered event detection that is to detect trigger words and assign correct event types. Some advanced methods (Poon and Vanderwende, 2010; Riedel and McCallum, 2011; Li et al., 2013, 2014; Venugopal et al., 2014; Judea and Strube, 2016; Nguyen et al., 2016; Sha et al., 2018) tried to unify two sub-tasks, event detection and argument extraction, but all assumed that entity candidates were given in advance. A few studies attempted to fulfill all sub-tasks of EE jointly. Yang and Mitchell (2016) was the first work towards this goal but relied on handcrafted features. Later research (Nguyen and Nguyen, 2019) explored the joint modeling further by introducing neural networks but also retained many traditional lexical and syntactic features. Recently, Zheng et al. (2019a) formalized a new succinct task for EE without trigger words and proposed the first document-level end-to-end model, called Doc2EDAG. This novel model transformed the task of filling an event table into the generation of an EDAG and thus enabled the end-to-end modeling for EE.

With the rapid development of modeling techniques, labeled data gradually became the main bottleneck that prevented from putting EE into practice. Most of the previous research was based on ACE 2005[1], a benchmark dataset annotated by human experts. However, human annotation is both expensive and time-consuming. Recent research attempted to generate weakly labeled data by aligning event instances from knowledge bases to plain text and then assigning labels to matched samples. This strategy was originated from relation extraction (Mintz et al., 2009; Zheng et al., 2019b), but most existing EE models relied on trigger words to anchor an event mention. Accordingly, two types of research explorations emerged: one was to label trigger words with the help of extra linguistic resources (Chen et al., 2017) or predefined dictionaries (Yang et al., 2018), and the other was to remove the requirement of trigger words in modeling (Zeng et al., 2018; Liu et al., 2019; Zheng et al., 2019a). In this paper, we follow the no-trigger-words design because it can ease the labeling work of EE and thus generate large-scale data.

Different from all the related work, this paper focuses on the evaluation aspect and attempts to extend the scope of EE evaluation methods to better support the *event-as-a-whole* scenarios.

3 Preliminaries

In this section, we provide readers with necessary background to better understand our research.

3.1 Terminologies

We follow Yang and Mitchell (2016) to use a general "**entity**" notion that covers persons, dates, numbers, and so on for brevity. Next, let us recall the ground-truth event table in Figure 2 and clarify some widely used terminologies for EE. 1) An **event role**, corresponding to a column of an event table, is a basic semantic unit of an event type (e.g., "Role 1" is an event role). 2) An **event argument**, corresponding to an entry of an event table, refers to an entity that plays a specific event role (e.g., "E1" is an event argument). 3) An **event instance**, corresponding to a row of an event table, consists of a group of entities that jointly characterizes a specific incident (e.g., "[E1, E2, E4]" together forms an event instance).

3.2 Doc2EDAG

As discussed in the related work, Doc2EDAG enabled not only end-to-end modeling for EE but

[1] https://www.ldc.upenn.edu/collaborations/past-projects/ace

also simplified event labeling. Therefore, we adopt Doc2EDAG as the base end-to-end model and use its associated large-scale benchmark for validation.

Given a text document as the input, Doc2EDAG first extracts entities and encodes them with the document-level context. Then, for a triggered event type, Doc2EDAG generates an entity-based directed acyclic graph (EDAG) via a series of path-expanding sub-tasks. Each path-expanding sub-task is composed of a group of binary predictions (1: *expanding* or 0: *not*), where one prediction happens to one entity candidate.

The vanilla training of Doc2EDAG follows a given ground-truth EDAG and calculates corresponding losses for path-expanding sub-tasks. We refer to this kind of training as the maximum likelihood estimation (MLE) because its essential goal is to maximize the likelihood of ground-truth EDAGs. However, the vanilla MLE-based training cannot fit the newly proposed metrics with hard constraints.

4 New Evaluation Metrics

Similar to the initial *role-averaged* evaluation, the new evaluation starts from comparing a predicted event table with the ground-truth one. To be specific, they pick one predicted event instance and the most similar ground-truth one without replacement from corresponding event tables. The difference lies in the comparison of two picked event instances. Initially, the *role-averaged* evaluation collects evaluation statistics of TP, FP, and FN for each event role and averages them to calculate precision, recall, and F_1 scores for that event type. In contrast, the new evaluation considers an event instance as a whole and collects those statistics directly at the event level. Next, we illustrate the details of two new metrics on collecting TP, FP, and FN statistics when taking into consideration the *event-as-a-whole* requirement.

The NoFP Metric. An event instance that contains any FP argument belongs to an FP error at the event level. For an event instance that contains both FN arguments and TP arguments, we retain the statistics of FN and TP in proportion at the event level. For an event whose arguments are all correct, we count it as a TP event.

The NoFPFN Metric. We treat an event instance that contains either FP arguments or FN arguments as an FP error at the event level. Only for those event instances that are the same to the ground-truth ones, we treat them as TP events.

Note that the *NoFP* metric can be beneficial to risk-sensitive scenarios, such as finance, health, etc., where FP arguments of an event instance may cause disastrous effects. Moreover, the *NoFPFN* metric, also the most challenging one, can tell us how large the gap between the current progress and the perfect extractor is. In contrast, the initial *role-averaged* evaluation can only provide the approximate measures by averaging the performance of event role predictions, which only fit for the cases when downstream applications utilize arguments of an event instance independently.

5 Optimizing New Metrics

Optimizing these new metrics is pretty challenging. First, these metrics are non-differentiable, so it is hard to specify corresponding training objectives. Besides, we can only calculate event-level metrics after obtaining predictions for all event roles, so the supervision for role-level predictions is inevitably delayed. Moreover, EDAG generation is an autoregressive procedure inherently, which suffers from the error-propagation problem.

To address these challenges, we develop an RL-based training paradigm for Doc2EDAG by viewing EDAG generation as a sequential decision-making process. In this way, we can explicitly optimize model parameters by a delayed reward, which can be specified by a non-differentiable metric. Moreover, the explorations during RL-based training also help to stabilize EDAG generation.

MDP. Instead of treating the EDAG generation as an autoregressive procedure, we regard it as a Markov decision process (MDP). First, we consider the event triggering as the beginning of this MDP, which is also the virtual starting node of the EDAG (Zheng et al., 2019a). Then, for a specific event role, which corresponds to one step of the MDP, we need to take specific actions for all encoded entities at all leaf nodes of the current EDAG. Next, the EDAG grows accordingly, and we move to the next event role. This iterative process continues until we reach the last role. By following specific evaluation metrics to compare the generated EDAG with the ground-truth one, we can obtain the final reward to guide the decision-making process from a global perspective.

State. In general, the state at each step is the part of EDAG before the current event role. In this

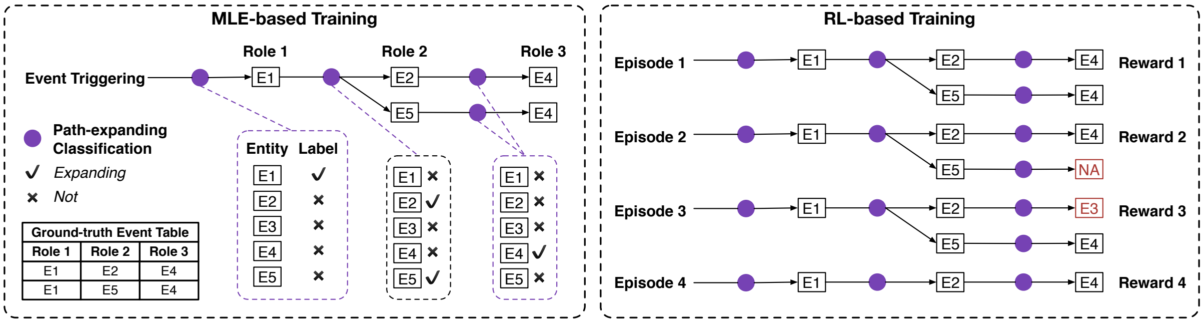


Figure 3: An overview of the comparison between MLE-based training and RL-based training for Doc2EDAG, where all marks follow the meanings in Figure 1 and 2.

paper, to present a fair comparison with the vanilla Doc2EDAG with MLE-based training, we leverage the same method developed by Zheng et al. (2019a) to obtain state representations of the current EDAG. In this way, we can focus on the real impact of different supervision paradigms. For the step t, we denote its state as $\boldsymbol{s}_t$.

Action. At each step of the MDP, we need to take an action, *expanding* (1) or *not* (0), for each entity candidate at every leaf node of the current EDAG. We denote the action collection at step t as $\boldsymbol{a}_t$. Here the action concept is the same as the path-expanding prediction used in MLE-based training. Note that there are multiple actions to be made (one action per entity per expanding sub-task), which is a little bit similar to the setting of multi-agent RL (Buşoniu et al., 2010). While the difference is that our model unifies all these agents in a single model and enforces them to work collaboratively for a specific metric. Moreover, we need to face a growing number of actions during the expansion of an EDAG, which largely raises the difficulties of model training. Despite these challenges, we empirically demonstrate that the RL-based training can work pretty well with proper configurations.

Reward. For a specific evaluation metric, we utilize its rules to calculate the reward and use this reward to guide the optimization of model parameters. Moreover, there is no gap between training and inference because the model generates an EDAG by itself to get a reward during training, which includes the inference procedure.

Optimization. Given a document d and a triggered event type e with N_e event roles, we can take a series of actions to get an episode $\tau = (\boldsymbol{s}_1, \boldsymbol{a}_1, \boldsymbol{s}_2, \boldsymbol{a}_2, \cdots, \boldsymbol{s}_{N_e}, \boldsymbol{a}_{N_e})$ and calculate the final reward R_τ by comparing the generated EDAG with the ground truth. Then, we can write the overall loss function for the EDAG generation as

$$L_{edag} = -\mathbb{E}_{d,e}\left[\mathbb{E}_{\pi_\Theta(\tau)} R_\tau\right], \qquad (1)$$

where $\mathbb{E}_{d,e}$ is the expectation over training documents and their triggered event types, Θ denotes the model parameters, $\pi_\Theta(\tau) = \prod_{t=1}^{N_e} \pi_\Theta(\boldsymbol{a}_t|\boldsymbol{s}_t)$ estimates the probability of an episode, and $\pi_\Theta(\boldsymbol{a}_t|\boldsymbol{s}_t)$ corresponds to the policy at step t. We employ the REINFORCE algorithm (Williams, 1992) and the policy gradients (Sutton et al., 1999) to optimize the above objective. And we can write the gradient $\nabla_\Theta L_{edag}$ as

$$-\mathbb{E}_{d,e}\left[\mathbb{E}_{\pi_\Theta(\tau)}\left[\sum_{t=1}^{N_e} R_\tau \log \pi_\Theta(\boldsymbol{a}_t|\boldsymbol{s}_t)\right]\right]. \qquad (2)$$

Moreover, we also follow Zheng et al. (2019a) to obtain the final loss by summing L_{edag}, the loss of entity recognition L_{er}, and the loss of event triggering L_{et} as $L_{all} = \lambda_1 L_{er} + \lambda_2 L_{et} + \lambda_3 L_{edag}$, where λ_1, λ_2 and λ_3 are hyper-parameters.

Exploration. In our case, the complexity of generating an EDAG grows exponentially with the number of entities and path-expanding sub-tasks. To achieve efficient explorations under such a challenging scenario, we start from MLE-based training to get a relatively well-trained model and warm start RL-based training from it. After the warm start, we let the model sample actions according to the predicted path-expanding probability to achieve proper explorations.

Figure 3 depicts the comparison between MLE-based training and RL-based training, where we can observe two benefits of the latter one: 1) the model learns from the global supervision (reward), which can be specified by a specific metric; 2) the

Event	#Train	#Dev	#Test	#Total
EF	806	186	204	1,196
ER	1,862	297	282	3,677
EU	5,268	677	346	5,847
EO	5,101	570	1,138	6,017
EP	12,857	1,491	1,254	15,602
All	25,632	3,204	3,204	32,040

Table 1: Statistics of the ChFinAnn dataset, including the number of documents on the train (#Train), development (#Dev), and test (#Test) sets as well as the total number of documents (#Total).

model can explore diverse EDAG structures during training, which helps to stabilize the dynamic process of EDAG generation.

6 Experiments

In this section, we present extensive empirical studies to answer two questions: 1) how severe is the overestimation of the initial *role-averaged* evaluation under the *event-as-a-whole* requirement? 2) to what extent can RL-based training improve MLE-based training under new evaluation metrics? Subsequently, Section 6.2 answers the first question. Section 6.3, 6.4, and 6.5 together answer the second question.

6.1 Experimental Setup

Model. As Section 3.2 illustrates, Doc2EDAG owns the superiority of both providing end-to-end modeling and simplifying data labeling. Therefore, in this paper, we adopt Doc2EDAG as the base model to demonstrate the challenges of new evaluation settings and the benefits of the RL-based training paradigm. Moreover, we follow the open-source implementation[2] to reproduce MLE-based training.

Dataset. The ChFinAnn dataset (Zheng et al., 2019a) is the largest public dataset for EE at present, and it is also the initial benchmark for Doc2EDAG. Therefore, we employ it as the testbed to compare RL-based training with the initial MLE-based training. ChFinAnn dataset includes ten-years Chinese financial documents accompanied with corresponding event tables and contains five event types: *equity freeze* (EF), *equity repurchase* (ER), *equity underweight* (EU), *equity overweight* (EO), and *equity pledge* (EP). Table 1 summarizes this dataset.

Metric	Avg. P.	Avg. R.	Avg. F_1
Base*	-	-	76.3
Base	83.8	70.6	76.6
NoFP	60.4	52.4	56.1
Drop	-23.4	-18.2	-20.5
NoFPFN	37.5	34.8	36.1
Drop	-46.3	-35.8	-40.5

Table 2: We present the overall performance (Avg.) of the vanilla Doc2EDAG under different evaluation settings by averaging scores for each event type, where * indicates the copy of results in Zheng et al. (2019a),"Base" denotes the initial *role-averaged* evaluation, "NoFP" denotes the *NoFP* evaluation, "NoFPFN" denotes the *NoFPFN* evaluation, and "Drop" represents the difference between the line above it and the "Base" line.

Hyper-parameters. MLE-based training sets all hyper-parameters the same as those presented in Zheng et al. (2019a). As for RL-based training, we utilize the following settings. To enable efficient explorations, we first train Doc2EDAG by MLE for 30 epochs and then turn to RL for another 70 epochs, where the total number of training epochs is still 100. To alleviate the noises of policy gradients, we sample five independent episodes for each document in the training batch and adopt a relatively small learning rate of $1e^{-5}$. For all other hyper-parameters, we set them the same as those in MLE-based training. Moreover, for each training method, we utilize the development set to pick the best epoch for the target evaluation metric.

Evaluation. We consider three evaluation settings: 1) the *role-averaged* evaluation, which is the base setting that provides approximate measures by averaging the performance of event roles; 2) the *NoFP* evaluation, strictly penalizing incorrect event arguments; 3) the *NoFPFN* evaluation, explicitly penalizing both incorrect and incomplete event arguments. For all settings, we collect evaluation statistics of TP, FP, and FN as Section 4 describes to compute the precision (P.), recall (R.), and F_1 scores (in the percentage format).

6.2 The Overestimation Problem of the Role-averaged Evaluation

In this section, we conduct experiments to reveal the overestimation problem of the initial *role-averaged* evaluation under the *event-as-a-whole* scenario. Following the setups mentioned above,

[2] https://github.com/dolphin-zs/Doc2EDAG

Metric	EF			ER			EU			EO			EP		
	P.	R.	F_1	P.	R.	F_1	P.	R.	F_1	P.	R.	F_1	P.	R.	F_1
Base*	77.1	64.5	70.2	91.3	83.6	87.3	80.2	65.0	71.8	82.1	69.0	75.0	80.0	74.8	77.3
Base	78.4	66.5	71.9	95.5	81.3	87.9	80.4	62.0	70.0	80.7	70.9	75.5	84.2	72.4	77.9
NoFP	49.7	46.3	47.9	78.1	70.6	74.2	61.7	48.5	54.3	58.7	48.0	52.8	53.7	48.7	51.1
Drop	-28.7	-20.2	-24.0	-17.4	-10.7	-13.7	-18.7	-13.5	-15.7	-22.0	-22.9	-22.7	-30.5	-23.7	-26.8
NoFPFN	36.2	32.1	34.0	40.1	39.4	39.8	39.2	35.6	37.3	35.2	33.3	34.3	36.5	33.6	35.0
Drop	-42.2	-34.4	-37.9	-55.4	-41.9	-48.1	-41.2	-26.4	-32.7	-45.5	-37.6	-41.2	-47.7	-38.8	-42.9

Table 3: We evaluate the vanilla Doc2EDAG under different settings and report results for all event types, where all marks follow those in Table 2.

we reproduce a Doc2EDAG model with the initial MLE-based training. Then, we evaluate this model with different evaluation methods. Table 2 presents the overall comparison, and Table 3 records performance comparisons for each event type. We can observe that the performance results of our reproduced model roughly match those reported by Zheng et al. (2019a) under the base setting. However, there is a drastic drop in the performance scores under new metrics. On average, the F_1 score decreases 20.5 under the *NoFP* metric, and the F_1 decrement reaches 40.5 under the *NoFPFN* metric. The *role-averaged* metric does not consider the *event-as-a-whole* requirement and tends to produce overestimated performance that is too optimistic to fit the real situation. The vast performance degradation demonstrates that the overestimation problem can be quite severe.

Moreover, the degree of overestimation depends on the event type and varies irregularly. For example, as Table 3 shows, the model achieves similar F_1 scores for EF and EU events (71.9 vs. 70.0), but the performance drops under the *NoFP* evaluation are quite different (-24.0 vs. -15.7). The underlying reason is that the *role-averaged* evaluation only tells us the averaged performance of role predictions, which is an approximate measure. Therefore, in downstream applications that utilize an event instance as a whole, it is critical for them to utilize our newly designed metrics to be aware of the real performance.

6.3 The NoFP Evaluation

The *NoFP* evaluation considers an event instance as a whole and strictly penalizes FP entities. Under this evaluation setting, we compare the RL-based training that directly optimizes the specific metric with the initial MLE-based training that mimics path-expanding predictions of a ground-truth EDAG. Table 4 shows the overall performance comparisons, and Figure 4 presents performance improvements of RL-based training for all event types. We can observe that, on average, RL-based training improves the MLE-based one by 8.9% of F_1 scores, and similar improvements happen for all event types. These vast improvements demonstrate that considering the *NoFP* metric explicitly during training can bring remarkable benefits.

Model	Avg.		
	P.	R.	F_1
MLE	60.4	52.4	56.1
RL	66.2	56.7	61.1
Inc	+5.8	+4.3	+5.0

Table 4: The averaged (Avg.) performance of all event types under the *NoFP* evaluation, where "MLE" denotes Doc2EDAG with MLE-based training, "RL" denotes the model with RL-based training, and "Inc" represents the increment between "RL" and "MLE".

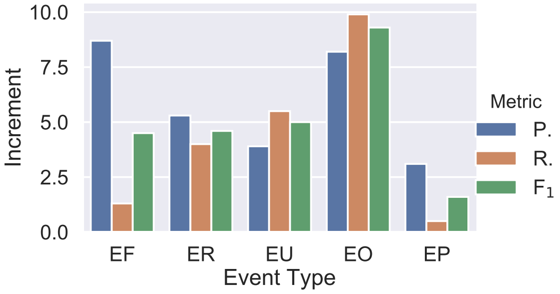


Figure 4: The performance increment between RL-based training and MLE-based training for all event types under the *NoFP* evaluation.

6.4 The NoFPFN Evaluation

We further consider the *NoFPFN* evaluation that penalizes not only FP entities but also FN entities. As Table 5 shows, RL-based training obtains similar improvements over MLE-based training on average. To be specific, the relative improvement in terms of the F_1 score reaches 11.3%. Since RL-

Model	Avg.		
	P.	**R.**	**F$_1$**
MLE	37.5	34.8	36.1
RL	39.6	40.7	40.2
Inc	+2.1	+5.9	+4.1

Table 5: The averaged (Avg.) performance of all event types under the *NoFPFN* evaluation, where all abbreviations follow Table 4.

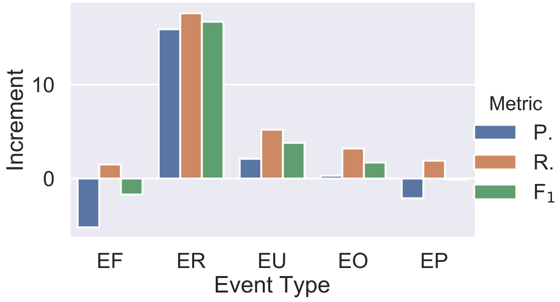


Figure 5: The performance increment between RL-based training and MLE-based training for all event types under the *NoFPFN* evaluation.

based training and MLE-based training share the same network architecture, we can clearly see the benefits of evaluation-aware learning.

Moreover, Figure 5 shows performance improvements of RL-based training for all event types. Interestingly, under the much challenging *NoFPFN* evaluation, we observe that the model automatically finds a trade-off between different event types. For example, the model sacrifices a little performance on EF (-1.7) and EP (-0.1) events but earns much more improvements on ER ($+16.7$), EU ($+3.8$), and EO ($+1.7$) events. Such an automatic trade-off is quite appealing because it is entirely data-driven and can adapt to different scenarios flexibly.

6.5 The Role-averaged Evaluation

In this subsection, we include performance comparisons under the initial setting, the *role-averaged* evaluation, where RL-based training follows the initial rule to calculate rewards. Table 6 shows the averaged comparison results. We can observe that MLE-based training achieves competitive performance compared with RL-based training. The underlying reason is that both MLE-based training and the *role-averaged* evaluation focus on improving the role-level predictions, so these two techniques are much consistent with each other. Nevertheless, RL-based training still obtains a small gain of the F$_1$ score ($+1.1$) because it can make a proper trade-off between the precision and the recall. Figure 6 presents performance improvements for all event types, and we also observe similar trade-offs. Therefore, we can conclude that RL-based training can achieve remarkable improvements under new metrics while still maintain competitive performance for the initial setting.

Model	Avg.		
	P.	**R.**	**F$_1$**
MLE	83.8	70.6	76.6
RL	79.9	75.7	77.7
Inc	-3.9	+5.1	+1.1

Table 6: The averaged (Avg.) performance of all event types under the *role-averaged* evaluation, where all abbreviations follow Table 4.

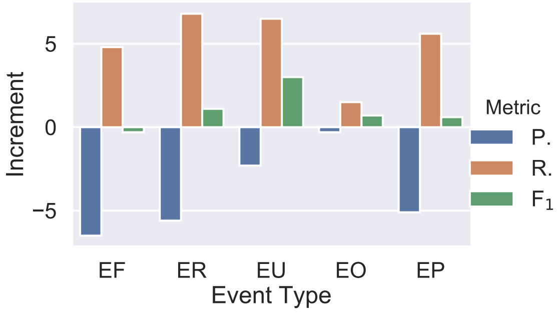


Figure 6: The performance increment between RL-based training and MLE-based training for all event types under the *role-averaged* evaluation.

7 Conclusion and Future Work

In conclusion, our new metrics provide a more complete understanding of performance than the *role-averaged* metric under the *event-as-a-whole* scenario. Moreover, our proposed RL-based training for Doc2EDAG can be a better choice than the MLE-based one when adapting the model to these new metrics. Extensive experiments demonstrate the necessity of new evaluation metrics and the superiority of the novel training scheme.

Given the necessity of employing new metrics, however, as Table 2 shows, the current performance is still far from satisfactory, even for RL-based training. Therefore, further improving EE under these new metrics calls for much more attention from the research community.

Acknowledgements

This research is supported in part by gift funds from the Nanjing Turing AI Institute and the Zhongguancun Haihua Institute for Frontier Information Technology.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning About Time and Events*.

Lucian Buşoniu, Robert Babuška, and Bart De Schutter. 2010. Multi-agent reinforcement learning: An overview. In *Proceedings of Innovations in multi-agent systems and applications-1*. Springer.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *Proceedings of ACL*.

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of EMNLP*.

Alex Judea and Michael Strube. 2016. Incremental global event extraction. In *Proceedings of COLING*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of ACL*.

Qi Li, Heng Ji, HONG Yu, and Sujian Li. 2014. Constructing information networks using one single model. In *Proceedings of EMNLP*.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of ACL*.

Shulin Liu, Yang Li, Feng Zhang, Tao Yang, and Xinpeng Zhou. 2019. Event detection without triggers. In *Proceedings of NAACL*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL*.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of NAACL*.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of ACL*.

Trung Minh Nguyen and Thien Huu Nguyen. 2019. One for all: Neural joint modeling of entities and events. In *Proceedings of AAAI*.

Hoifung Poon and Lucy Vanderwende. 2010. Joint inference for knowledge extraction from biomedical literature. In *Proceedings of NAACL*.

Sebastian Riedel and Andrew McCallum. 2011. Fast and robust joint models for biomedical event extraction. In *Proceedings of EMNLP*.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction. In *Proceedings of AAAI*.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of NIPS*.

Deepak Venugopal, Chen Chen, Vibhav Gogate, and Vincent Ng. 2014. Relieving the computational bottleneck: Joint inference for event extraction with high-dimensional features. In *Proceedings of EMNLP*.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019. Adversarial training for weakly supervised event detection. In *Proceedings of NAACL*.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of NAACL*.

Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, and Jun Zhao. 2018. DCFEE: A document-level Chinese financial event extraction system based on automatically labeled training data. In *Proceedings of ACL 2018, System Demonstrations*.

Ying Zeng, Yansong Feng, Rong Ma, Zheng Wang, Rui Yan, Chongde Shi, and Dongyan Zhao. 2018. Scale up event extraction learning via automatic training data generation. In *Proceedings of AAAI*.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019a. Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction. In *Proceedings of EMNLP*.

Shun Zheng, Xu Han, Yankai Lin, Peilin Yu, Lu Chen, Ling Huang, Zhiyuan Liu, and Wei Xu. 2019b. DIAG-NRE: A neural pattern diagnosis framework for distantly supervised neural relation extraction. In *Proceedings of ACL*.