Transferring Knowledge using Gradient Similarity for Document-level Event Extraction

Anonymous EMNLP submission

Abstract

Document-level event extraction (DocEE) aims to extract structured event knowledge from unstructured document data. Previous DocEE works mainly focused on task modeling, ignoring the persistent low-resource issue (In this paper, it refers to the lack of annotated training data specifically). Therefore, we propose a novel method to solve this issue, which can implicitly learn the task knowledge required by DocEE from data of other information extraction (IE) tasks. We first construct auxiliary tasks using data of related IE tasks (e.g., NER, RE.) and then transfer their knowledge with a multi-task learning strategy. In addition, the adaptive knowledge transfer mechanism based on gradient similarities is introduced in the process of multi-task learning to alleviate the problem that auxiliary tasks may introduce too much noise and irrelevant knowledge. The experimental results on the MUC-4 dataset show that our method achieves state-of-the-art performance on DocEE (+5.51% in F1-score). Additional analysis shows that the pseudo data generated by IE tools can also be applied in our framework, demonstrating the practicability and robustness of our framework against noise.

1 Introduction

Event extraction (EE) is an important and challenging task in natural language processing. It targets to extract structured events of specific types from unstructured texts. Each type of event contains multiple argument roles. Figure 1 depicts an ATTACK event (the event type) accompanied by its arguments and their types. Most of the current studies focus on sentence-level event extraction (SentEE), where they assume that an event is represented in a sentence (Du and Cardie, 2020b; Liu et al., 2018). However, in most cases, one or more events may exist in multiple non-contiguous sentences. In this work, we mainly focus on document-level event extraction (DocEE).

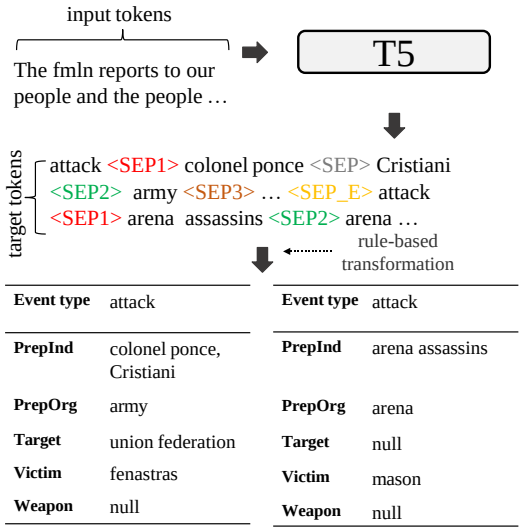

Figure 1: Sequence to sequence generation for document-level event extraction. Given an input document, T5 decodes the corresponding sequence of events and then translates them into structured templates by rules.

One long-recognized issue with DocEE is the lack of annotated training data (Yang et al., 2018). A possible solution to this problem is to leverage transfer learning techniques (Ruder et al., 2019; Vu et al., 2020). Existing works on transfer learning for EE aim to transfer knowledge from related tasks and use pre-trained language models (PLMs) like BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020). For example, Lu et al. (2022) design transferable neural architectures based on PLMs to perform transfer learning for SentEE. In contrast, other works transfer knowledge from similar tasks (Lyu et al., 2021; Liu et al., 2021; Van Nguyen et al., 2021) or use related tasks to assist the model with inferencing (Lin et al., 2020). Although these works have made some progress, they still do not make use of all transferable knowledge.

Recently, some works utilize both PLMs and

similar tasks to perform transfer learning (Paolini et al., 2021; Lu et al., 2022). For example, TANL (Paolini et al., 2021) formulates multiple IE tasks as similar forms, then uses T5 to complete all tasks simultaneously by multi-task learning and allowing knowledge transfer between similar tasks. However, these efforts tend to transfer knowledge from other tasks in their gold forms, which has two main drawbacks. First, gold forms of other tasks contain irrelevant knowledge to the target task. As a result, the model may not be able to learn the representation that can be generalized in tasks in the learning process (Fifty et al., 2021). Additionally, there may be noise in other tasks, which will lead to a negative transfer.

In this work, we are motivated to address these drawbacks and propose an effective multi-task learning framework to perform transfer learning. Compared with previous works, we reduce the transfer of irrelevant knowledge by formulating other tasks into forms more conducive to transfer rather than using gold forms directly. Meanwhile, to further reduce the negative transfer, we propose the adaptive knowledge transfer mechanism based on gradient similarities (Wang et al., 2020; Li and Gong, 2021). Specifically, according to the characteristics of PLMs (Jawahar et al., 2019; Rogers et al., 2020), we discover and alleviate the transferred noise and irrelevant knowledge at the layer level of PLMs. In the implementation, we first perform a round of naive multi-task learning to calculate gradient similarity in different layers. And then, when performing multi-task learning for knowledge transfer, we only share gradients between tasks in layers with high similarity to alleviate the negative transfer. To evaluate our approach, we conduct experiments in the MUC-4 dataset (muc, 1992) for DocEE. Compared with baselines, our framework outperforms them to achieve a new state-of-the-art (+5.51% in F1-score).

In summary, our contributions are as follows:

- We propose a method to transfer knowledge to DocEE using the multi-task learning strategy for the first time. The strategy employs other related IE tasks as auxiliary tasks and transfers knowledge implicitly.
- We use gradient similarities to discover the noise and irrelevant knowledge of auxiliary tasks in the process of multi-task learning and propose the adaptive transfer mechanism to reduce the impact of noise or irrelevant knowledge.
- Our experimental results show that our method significantly improves the performance of the baseline. Compared with previous methods, our method achieves a new state-of-the-art performance.
- We also investigate the availability of our method when we construct auxiliary tasks by pseudo data. Experimental results show that our framework is robust to noise from pseudo data and prove it has good practicability in real scenarios.

2 Related Work

2.1 Document-Level Event Extraction

Previous works of document-level event extraction (DocEE) mainly focused on end-to-end DocEE or document-level role-filler entity extraction (DocREE). For example, Du and Cardie (2020a) used multi-granularity contextualized encoding methods to enhance their sequence tagging model. In contrast, Du et al. (2021a) formulated DocREE as a sequence-to-sequence task and generated entity sequences with an encoder and decoder that share parameters. Their experiments showed that the generative structure can implicitly capture the co-referential relationship between entity mentions. Huang et al. (2021) further found that label semantics can enhance a sequence-to-sequence model generation performance in entity-based generative extraction tasks. Besides, end-to-end DocEE models have also been explored. For example, Du et al. (2021b) based on their previous work and formulated DocEE as an end-to-end generation task by subtly modifying model inputs and outputs.

In the Chinese financial DocEE task, the input text is usually long and the cost of annotating data is expensive. In this application scenario, DCFEE (Yang et al., 2018) proposed a method to automatically generate large-scale data and extract events from the entire document. In contrast, Doc2EDAG (Zheng et al., 2019) implemented a DocEE system effectively by generating an entity-based directed acyclic graph. And Xu et al. (2021) built a heterogeneous graph-based interaction model with a tracker to perform DocEE.

Previous work mainly focused on DocEE at the model level, trying to solve the existing issues of DocEE with a better model. However, our work aims to transfer knowledge of auxiliary tasks to

DocEE, which can further improve DocEE performance in low-resource scenarios.

2.2 Multi-task Learning in IE

Recently, multi-task learning in IE has attracted extensive attention. These works can be classified into two main categories. The first category of works perceives that the previous partition of IE subtasks or pipeline systems will make the model lose the ability to obtain the globally optimal result (Yang and Mitchell, 2016; Liu et al., 2018). Therefore, merging multiple subtasks into a single task is necessary to achieve the globally optimal result. For example, OneIE (Lin et al., 2020) believes that named entity recognition (NER), relation extraction (RE), and SentEE are related tasks centered on entities. Considering their correlation, OneIE merges them into a graph generation task and obtains an IE graph close to globally optimal through beam search.

The second category of works deems that if they can formulate multiple IE tasks as formally unified tasks then they can use a unified framework to solve these IE tasks simultaneously by using multi-task learning (Lu et al., 2022; Paolini et al., 2021). Meanwhile, in this category of work, different tasks are not merged. For example, TANL (Paolini et al., 2021) formulated multiple sentence-level IE tasks in a unified form. The unified task form enables the model to have good multi-task learning ability. Therefore, it can solve many tasks with a single model. Experimental results show that it achieves comparable performance with the performance of training a single task with a single model in both multi-task learning and multi-dataset settings.

However, our work is different from either. Inspired by the similarity between document-level IE tasks, we propose that we can transfer the knowledge of other IE tasks to DocEE implicitly through multi-task learning. Differing from the second category of tasks, we do not need to formulate all tasks in a unified form (which is also hard to formulate). Meanwhile, we also pay more attention to the performance of DocEE, while auxiliary tasks only provide the role of knowledge transfer in the training phase. Furthermore, we achieved better performance on DocEE by controlling the optimization process of the model.

3 Preliminaries

3.1 Task Formalization

Suppose there are m event types $\{T_1, T_2, ..., T_m\}$ in the dataset. Each event contains n event roles $\{r_1, r_2, ..., r_n\}$. For each event role, there may be k entities, and k is not a constant number, but is determined by the specific event instances. For each document D_i containing l words $\{x_1, x_2, ..., x_l\}$, we need to extract all the event instances contained in its text (the number of event instances can be zero). For each event instance E_i, we need to assign it a corresponding event type T_i and extract all the entities $\{e_1, e_2, ..., e_k\}$ corresponding to each event role (the number of entities can be zero). In this paper, we assume there is a general template across all event types.

3.2 DocEE as Sequence Generation

We use the generative event extraction method defined and used by GTT (Du et al., 2021b) and Text2Event (Lu et al., 2021) as the baseline for our multi-task learning framework. This is mainly because in DocEE, the generative method achieves SOTA performance. Moreover, it is easier to implement multi-task learning using sequence to sequence structure than discriminant structure or sequence tagging structure. By using sequence to sequence structure as the backbone model, we don't need to modify the model to implement task-specific classifier and just need to focus on defining task-specific inputs and outputs.

As shown in Figure 1, we formalize DocEE in the following form: For each input document D_i, we need to generate its corresponding event sequence O_i. O_i consists of words contained in D_i and task-specific special tags in S, where S = {<SEP>, <SEP_E>, <SEP_1>, <SEP_2>,...<SEP_5>}. These special tags serve to separate generated event roles r_i and prompt the model during the generation process. For multiple event instances, we use <SEP_E> to separate different event instances. For each event instance E_i, we use different kinds of separators to separate different event roles r_i (e.g., <SEP_3> for $target$, <SEP_4> for $victim$), unlike the previous GTT template. Because we found that if we use GTT template as the decoded target, the decoder may not be aware of the class of r_i it needs to generate due to the use of the same separators, resulting in generated entities e_i being assigned to the wrong event role type.

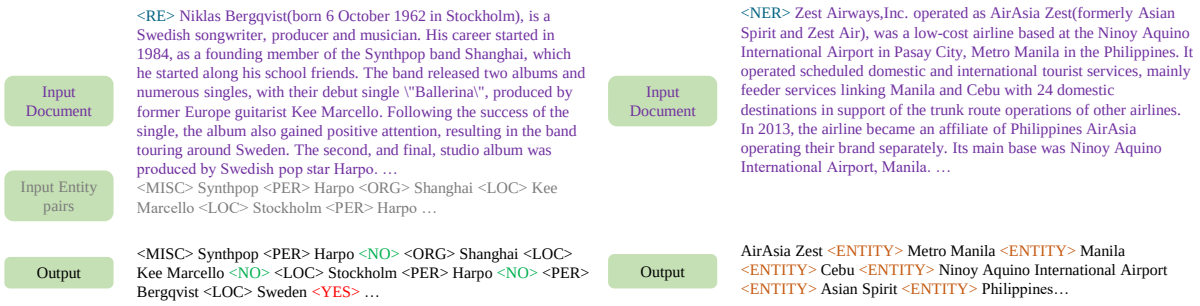

Figure 2: The diagram on the left is the task form of RE. Given the input document and the sequence of entity pairs, the model needs to determine whether the entity pairs have a relation (<YES> or <NO>). The diagram on the right is the task form of NER. Given the input document, the model should output the corresponding entity sequence.

Furthermore, we do not use the template defined by Text2Event as the decoded target, since tree structures are complex. Meanwhile, we can modify it to a linear form without losing the ability to represent event instances in the DocEE task. In addition, because of the complexity of the tree structure template, the model needs to introduce constraint decoding (Lu et al., 2021). Otherwise, there will be a certain probability for the model to generate a wrong sequence that we can not transform into a structured template. (e.g., generated open parenthesis has no corresponding close parenthesis). We conduct an ablation study about different templates in Table 4.

4 Proposed Method

In this section, we will cover our PMTL-AKT framework in more detail. PMTL-AKT framework is mainly composed of three parts: (1) construction of auxiliary tasks, (2)pseudo multi-task learning (PMTL) and (3) adaptive knowledge transfer (AKT). Figure 3 shows the main flow of PMTL-AKT. Firstly, traditional document-level IE tasks related to DocEE should be constructed as auxiliary tasks that are easier to transfer knowledge. Secondly, we use T5 (Raffel et al., 2020) as the backbone of our model, and use gradient similarities (Wang et al., 2020) to quantify the correlation between the DocEE task and auxiliary tasks at different layers of T5. Finally, we use the PMTL-AKT framework to make the model learn the DocEE task and transfer the knowledge of auxiliary tasks simultaneously.

4.1 NER and RE as Auxiliary Tasks

We select existing tasks (NER and RE) related to DocEE in document-level IE as auxiliary tasks. NER is a suitable auxiliary task because DocEE needs the knowledge of entity recognition it contains. Meanwhile, RE is a proper auxiliary task since it contains the relational knowledge, which is contained in DocEE implicitly. (e.g., criminal individual and criminal organization in DocEE will also represent as PER-ORG relation in RE.)

However, auxiliary tasks also contain less relevant knowledge to DocEE. For example, since we use the NER data from a dataset of a different domain (We can not find NER data in the same domain), the types and distribution of entities in NER do not exactly agree with DocEE. Therefore, we do not suggest using auxiliary task data in its gold form directly for transfer learning, which may introduce irrelevant knowledge and result in suboptimal performance. Thus, we changed the decoded target of NER from sequence tagging to document-level entity extraction, making it more related to DocEE's task form. Meanwhile, since the entity type distributions of DocEE and NER are different, we did not take entity type in NER as our decoded target.

Equally, the distribution of the relation types of RE also can not generate a mapping with the implicit entity relation distribution of DocEE. Therefore, we transform the original RE task form by adding entity pairs in the input and removing relation types in the target sequence. In the new task form, the model only needs to classify whether every given entity pair has a relation or not. In addition, adding entity pairs in the input decouples the knowledge of NER and RE, which is helpful for us in studying the impact of relational knowledge and entity knowledge on transfer learning.

Figure 2 describes the formulation of auxiliary tasks. We conduct an ablation study in table 5 to prove that transformation of the task form of auxiliary tasks is beneficial for knowledge transfer.

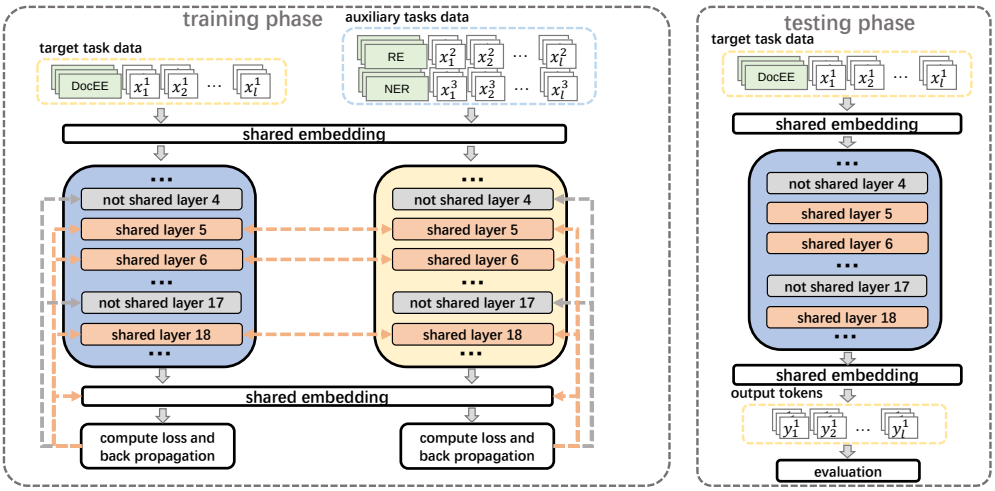

Figure 3: Our PMTL-AKT framework. The orange arrows show gradients shared between the target and auxiliary tasks, while the gray arrows stand for not shared gradients. We only left the shared embedding and layers of the target task for inference.

4.2 Pseudo Multi-task Learning Strategy

Our PMTL strategy is similar to Raffel et al. (2020) and TANL (Paolini et al., 2021). During the training and testing phase, to make the model recognize the current task $\mathcal{T} \in \{\mathcal{T}_i\}$, we simply attach the prefix token $\tau \in \{$<DocEE>, <NER>, <RE>$\}$ before the input tokens D. This method is simple yet effective, supported by experimental results (Paolini et al., 2021; Raffel et al., 2020). In the training phase, for all tasks, we simply adopt the training objective of sequence to sequence, that is, maximizing the probability of the target token y_j. Therefore, the loss function is

$$\mathcal{L} = \frac{1}{l} \sum_{i=1}^{l} -\log P(y_i),$$

where l is the length of a target sequence. In the testing phase, we use a greedy decoding strategy. At time step t, the model decodes y_t that maximizes the probability of the current sequence,

$$\max_{y_t} P(y_t | y_1 y_2 ... y_{t-1}).$$

For PMTL, the final loss function is the sum of the losses of all tasks,

$$\mathcal{L} = \sum_{i=1}^{|\{\mathcal{T}_i\}|} \mathcal{L}_i.$$

4.3 Use Gradient to Discover Noise and Irrelevant Knowledge in PMTL

Recent studies have found that PLMs tend to learn different knowledge in different layers (Jawahar et al., 2019; Rogers et al., 2020). Meanwhile, some works found that when applying PLMs to multi-task learning, knowledge fusion between tasks at specific layers is better than at all layers (Fang et al., 2021). Therefore, if PMTL strategy is adopted directly, auxiliary tasks will transfer more irrelevant knowledge and noise at these layers and affect the model's performance on DocEE (Fifty et al., 2021).

Inspired by previous multi-task learning optimization work (Wang et al., 2020; Li and Gong, 2021), we use gradient similarities to quantify the relevance between tasks. In the implementation, we adopted the following procedure: First, we initialize a gradient accumulation counter A_i for each task $\mathcal{T}_i$ and train DocEE and auxiliary tasks in the PMTL strategy. In the training phase, to ensure generated gradients are correctly recognized, a batch only contains data of a same task. Whenever the model performs back-propagation to calculate the gradient, we will distinguish the gradient generated by which task according to the prefix token in the input text. Then we add the gradient G_i to A_i. At the end of the training, for each layer L_k, we calculated the gradient similarities S_{ijk} between task i and task j according to A_{ik} and A_{jk},

$$S_{ijk} = \frac{A_{ik} \cdot A_{jk}}{\|A_{ik}\| \|A_{jk}\|}.$$

Based on the findings of multi-task learning optimization works (Wang et al., 2020), we deem that the lower the S_{ijk}, the less related $\mathcal{T}_i$ and $\mathcal{T}_j$ are in layer L_k. And then, the effect of $\mathcal{T}_i$ on $\mathcal{T}_j$ is closer to noise (and vice versa).

4.4 Adaptive Knowledge Transfer

The AKT mechanism aims to reduce the transfer of irrelevant knowledge and noise in PMTL. In the previous section, we quantified the relevance between tasks at different PLM layers. AKT takes advantage of this quantified information to reduce the transfer of irrelevant knowledge by decoupling highly relevant gradients from low-relevant gradients at the layer level. As shown in Figure 3, at layers of highly-relevant, we share gradients between tasks; at layers of low relevance, we do not share gradients between tasks. Instead, we let the gradients of the DocEE task and auxiliary tasks optimize their respective parameters. We elaborate main flow of AKT in Algorithm 1:

Algorithm 1: Adaptive Knowledge Transfer

Input: DocEE Task $\mathcal{T}$, Auxiliary Tasks $\{\mathcal{T}_i\}$, DocEE T5 Layers $\mathcal{M} = \{\theta_k\}$, Auxiliary T5 Layers $\bar{\mathcal{M}} = \{\bar{\theta}_k\}$, Shared Embedding θ_e, Similarity Between T5 Layers $\{S_k\}$, Threshold $\hat{S}$

Output: Trained DocEE T5 Layers $\mathcal{M}^*$ and Shared Embeddings ${\theta_e}^*$

while *not converged* **do**
  Sample task $\hat{\mathcal{T}}$ from $\mathcal{T} \cup \{\mathcal{T}_i\}$;
  Sample batch $\mathcal{B}$ from $\hat{\mathcal{T}}$;
  if $\hat{\mathcal{T}} == \mathcal{T}$ **then**
    $\hat{\mathcal{M}} = \mathcal{M}$;
  else
    $\hat{\mathcal{M}} = \bar{\mathcal{M}}$;
  end
  foreach $\hat{\theta}_k \in \hat{\mathcal{M}}$ **do**
    Compute gradient $g_k = \nabla_{\hat{\theta}_k} \mathcal{L}(\mathcal{B})$;
    if $S_k \geq \hat{S}$ **then**
      Update θ_k and $\bar{\theta}_k$ with g_k;
    else
      Update $\hat{\theta}_k$ with g_k;
    end
  end
  Compute gradient $g_e = \nabla_{\theta_e} \mathcal{L}(\mathcal{B})$;
  Update θ_e with g_e;
end
$\mathcal{M}^* = \mathcal{M}$, ${\theta_e}^* = \theta_e$;
Output: $\mathcal{M}^*$, ${\theta_e}^*$

5 Experimental Setup

5.1 Dataset and Evaluation Metric

We conduct our experiments on the MUC-4 (muc, 1992) dataset for DocEE. The MUC-4 dataset constituted 1700 documents with corresponding event templates. Each document contains 403.27 tokens and 7.12 paragraphs on average. Following previous works (Du et al., 2021b), we use 1300 documents for training, 200 documents (TST1+TST2) for validation, and 200 documents (TST3+TST4) for testing. For fair competition, we evaluate the DocEE task on MUC-4 using the same metric as Du et al. (2021b). Specifically, we evaluate our framework using per-slot F1-score and micro-average metrics for template generation (precision, recall, and F1-score). We obtain the training data of auxiliary tasks from Yao et al. (2019), a document-level relation extraction dataset with a complete entity and relation annotation.

5.2 Baselines

SEQTAGGING (Du and Cardie, 2020a) is a BERT-based sequence tagging model. It uses a multi-granularity contextualized encoding for extracting the role-fillers entities (REE). Du et al. (2021b) implements a pipeline version of it for DocEE Task.
DYGIE++ (Wadden et al., 2019) is a span enumeration based extractive model for information extraction.
GRIT (Du et al., 2021a) shares BERT parameters between the encoder and the pointer network decoder, and is the SOTA system for the REE task. Du et al. (2021b) implements a pipeline version called **GRIT-pipeline** for solving DocEE Task.
GTT (Du et al., 2021b) is an end-to-end DocEE model. It uses BERT to initialize model parameters, and share parameters between the encoder and the pointer network decoder to complete the DocEE task. It is also the SOTA system for the DocEE task on MUC-4 dataset.

6 Experiments Result

Results of the full testset are shown in Table 1 and Table 2. Experimental results of Table 1 show that our framework performs better than previous methods in all slots. Some performance increase comes from our more appropriate template (the performance of our baseline has surpassed previous methods), and another performance increase comes from the knowledge transferred by our PMTL-AKT framework. We discovered that using NER as an auxiliary task greatly enhanced the model's performance in the entity slots. While using RE as an auxiliary task enhances the model's ability to organize events, resulting in performance improvements at all slots. Meanwhile, we report three important evaluation metrics of DocEE in Table 2 (precision, recall, and micro F1-score). The results suggest that our framework significantly outperforms previous works, with 4.12%, 6.05%, and 5.5% increase compared with previous SOTA GTT (Du et al.,

Model	EVENT TYPE	PERPIND	PERPORG	TARGET	VICTIM	WEAPON
SEQTAGGING (Du and Cardie, 2020a)	60.22	30.59	26.79	36.60	43.62	51.7
DYGIE++ (Wadden et al., 2019)	61.95	32.44	25.73	45.04	49.48	51.60
GRIT-pipeline (Du et al., 2021a)	62.28	38.40	35.36	36.30	54.97	53.45
GTT (Du et al., 2021b)	67.44	44.04	41.79	32.39	54.12	59.71
Baseline	68.10	43.50	40.84	39.71	64.70	56.52
PMTL-AKT (RE)	70.92	42.42	42.26	**46.02**	67.05	57.60
PMTL-AKT (NER)	70.22	43.38	**45.60**	44.25	**67.54**	58.73
PMTL-AKT (RE&NER)	**70.93**	**45.71**	44.76	45.08	65.04	**59.78**

Table 1: Results on the MUC-4 dataset. Per-slot F1-score.

Model	P	R	F1
SEQTAGGING	46.80	38.30	42.13
DYGIE++	67.90	36.33	45.79
GRIT-pipeline	63.88	37.56	47.31
GTT	61.69	42.36	50.23
Baseline	61.33	46.55	52.62
PMTL-AKT (RE)	64.05	48.05	54.90
PMTL-AKT (NER)	65.04	48.11	55.28
PMTL-AKT (RE&NER)	**65.81**	**48.41**	**55.74**

Table 2: Results on the MUC-4 dataset. P, R and F1 stand for precision, recall and micro F1-score respectively.

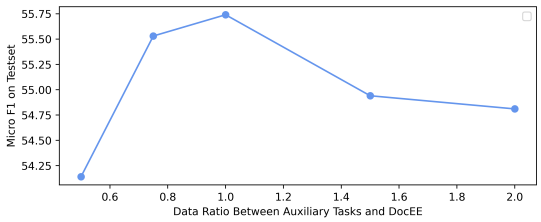

Figure 4: The impact of using different data ratio between auxiliary tasks and DocEE.

2021b) in precision, recall, and micro F1-score respectively.

Meanwhile, we also study how the data ratio of auxiliary tasks impacts the performance of DocEE. As shown in Figure 4, our PMTL-AKT framework works best when the data ratio between auxiliary tasks and DocEE is about 1:1.

6.1 Performance Analysis

The Impact of Pseudo Multi-task Learning

Table 2 gives the results of our baseline without PMTL. From the results, our PMTL strategy mainly improves the precision of our baseline. It suggests that the knowledge transferred by PMTL is more helpful in identifying false-positive samples. At the same time, it is also beneficial to improve the model's recall.

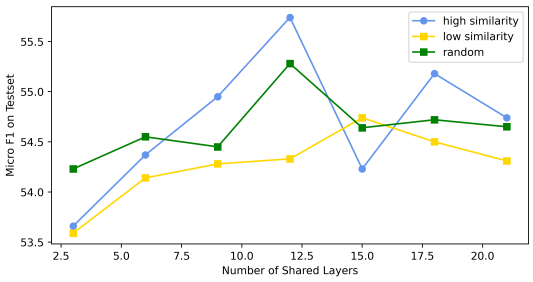

Figure 5: The effect of using different share strategies. Experimental results show that when sharing layers with high gradient similarities, the model performance is better than sharing layers with low gradient similarities or randomly assigning shared layers.

The Impact of Different Task Forms of Auxiliary Tasks

Table 5 gives the results of our framework, where gold forms of auxiliary tasks are used for transfer learning. Results show that when decoding target in NER contains entity type, model performance will drop 0.56% F1-score. Similarly, using the sequence of triples defined by Cabot and Navigli (2021)) as the decoding target of RE will lead to a performance decrease of 0.72% F1-score.

The Impact of Adaptive Knowledge Transfer

To analyze how the AKT mechanism impacts PMTL strategy, we report the results of our framework without an AKT mechanism in Table 3. We see that our framework's performance decreases both in precision and recall. Furthermore, our performance decrease comes from a high-level performance, which means our AKT mechanism can alleviate the impact of stubborn irrelevant knowledge and noise.

When implementing the AKT mechanism, the number of shared layers as a vital hyperparameter will seriously affect the model's performance. We study the correlation between it and the model's performance. We find that the model's performance is best when the number of shared layers is about

Model Variants	P	R	F1
PMTL-AKT (RE)	64.05	48.05	54.90
-AKT	61.37	47.63	53.53
PMTL-AKT (NER)	65.04	48.11	55.28
-AKT	65.09	46.68	54.29
PMTL-AKT (RE&NER)	65.81	48.41	55.74
-AKT	63.89	48.62	55.18
Average Δ	$\downarrow$ 1.52	$\downarrow$ 0.55	$\downarrow$ 0.97

Table 3: Results show that without the AKT mechanism, the model's performance decreased in both precision and recall. Average Δ: average decrease of metrics.

Template	DE	ECER
Entity type aware template	0%	3%
GTT template	0%	10%
Text2Event template	3%	5%

Table 4: The impact of using different event templates as the decoding target without constraint decoding. DE: decoding error rate. ECER: entity classifies error rate.

Model Variants	P	R	F1	Δ F1
PMTL-AKT (RE)	64.05	48.05	54.90	
with Gold Form	64.50	46.73	54.18	$\downarrow$ 0.72
PMTL-AKT (NER)	65.04	48.11	55.28	
with Gold Form	64.28	47.66	54.72	$\downarrow$ 0.56

Table 5: The impact of using the gold form of auxiliary tasks for our framework.

Training Strategy	P	R	F1
Baseline	61.50	46.01	52.51
PMTL-AKT (RE)	65.40	47.72	55.17
PMTL-AKT (NER)	67.00	45.89	54.47
PMTL-AKT (RE&NER)	64.27	48.96	55.57

Table 6: Results on the MUC-4 dataset, we use pseudo data for our framework.

half. As shown in Figure 5, When the number of shared layers is smaller than half, the model's performance will be decreased due to the inadequate introduction of auxiliary knowledge and is closer to the baseline. When the number of sharing layers of the model is more than half, the model will introduce more irrelevant knowledge and noise, and its performance is close to the baseline with PMTL.

The Impact of Using Gradient as Index

To prove that gradient similarities are a reasonable index to quantify the correlation between the target task and auxiliary tasks, we implement a reverse version of PMTL-AKT. In the reverse version, layers with low gradient similarities are shared between tasks (and vice versa).

Figure 5 shows that if we use the reverse version as the training strategy, the model will reduce its performance due to more irrelevant knowledge. We also examine the model's performance when shared layers are randomly assigned. We find that performance of the model is between high-similarity sharing and low-similarity sharing.

6.2 Other Analysis

When Our Framework Encounters Pseudo Data

For simulating more realistic application scenarios, we use existing information extraction tools (Manning et al., 2014) and document-level corpus (Yao et al., 2019) to generate some document-level pseudo data for our framework. To be specific, to control experimental variables, we generate pseudo data for NER and RE. Meanwhile, we transform the generated pseudo data into the task form consistent with the previous method of constructing auxiliary tasks. In the training phase, we used the same strategy described in Section 4. The experimental results are shown in Table 6. Results show that our PMTL-AKT framework performs well when using pseudo data, which proves our framework is robust to noise and practical in real scenarios.

Comparison Between Event Templates

Table 4 gives the results of using different event templates as the decoded targets. From the results, with entity type aware separator, the entities classify error rate of our template dropped significantly compared with the GTT template (Du et al., 2021b). Furthermore, since we simplified the template form, templates generated by our baseline could achieve 100% decoding accuracy even without using constraint decoding (Lu et al., 2021).

7 Conclusion

In this paper, we have proposed a new perspective to handle the lack of annotated training data of DocEE. We construct two kinds of auxiliary tasks for DocEE and use the pseudo multi-task learning strategy to transfer the knowledge to DocEE. In addition, we also introduce the adaptive knowledge transfer mechanism to alleviate the issue that auxiliary tasks may transfer too much noise or irrelevant knowledge. The extensive experiments on the MUC-4 dataset demonstrate the effectiveness of our approach. Finally, experimental results with pseudo data have justified our method's practicability and robustness.

Limitations

Since we only prove the effectiveness of our framework in the DocEE task, we need to verify the generalization ability of our framework in more scenarios. At the same time, our framework requires twice as many GPU resources to calculate gradient similarities compared with using multi-task learning directly. In addition, in the process of multi-task learning, auxiliary tasks will also bring the extra cost of computing resources. Finally, the performance of our framework under complex samples still needs to be improved. For example, when there are multiple events in the same document or multiple entities in the same event role, the model enhanced by our framework still performs poorly. There are two main reasons for this phenomenon. One is that such samples are complex and sparse; the other is that our constructed auxiliary tasks are simple. Therefore, building appropriate auxiliary tasks to enhance model performance on difficult samples is a vital research issue and should be discussed in the future.

References

1992. *Fourth Message Uunderstanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. Rebel: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381.

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.

Xinya Du and Claire Cardie. 2020a. Document-level event role filler extraction using multi-granularity contextualized encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8010–8020.

Xinya Du and Claire Cardie. 2020b. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683.

Xinya Du, Alexander M Rush, and Claire Cardie. 2021a. Grit: Generative role-filler transformers for document-level event entity extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 634–644.

Xinya Du, Alexander M Rush, and Claire Cardie. 2021b. Template filling with generative transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 909–914.

Yuwei Fang, Shuohang Wang, Zhe Gan, Siqi Sun, and Jingjing Liu. 2021. Filter: An enhanced fusion method for cross-lingual language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12776–12784.

Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. 2021. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, 34.

Kung-Hsiang Huang, Sam Tang, and Nanyun Peng. 2021. Document-level entity-based extraction as template generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5257–5269.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Xian Li and Hongyu Gong. 2021. Robust optimization for multilingual translation with imbalanced data. *Advances in Neural Information Processing Systems*, 34.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009.

Jian Liu, Yufeng Chen, and Jinan Xu. 2021. Machine reading comprehension as data augmentation: A case study on implicit event argument extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2716–2725.

Xiao Liu, Zhunchen Luo, and He-Yan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the*

11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 2795–2806.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. *arXiv preprint arXiv:2101.05779*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Sebastian Ruder, Matthew E Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials*, pages 15–18.

Minh Van Nguyen, Tuan Ngo Nguyen, Bonan Min, and Thien Huu Nguyen. 2021. Crosslingual transfer learning for relation and event extraction via word category and class alignments. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5414–5426.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across nlp tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7882–7926.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. 2020. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 2021. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3533–3546.

Bishan Yang and Tom Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299.

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*, pages 50–55.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. Docred: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2edag: An end-to-end document-level framework for chinese financial event extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 337–346.

A Implementation Details

All of our experiments are based on T5-base (Raffel et al., 2020). Hyperparameters search ranges are 1) learning rate: (1e-5, 5e-5, 1e-4); 2) batch size: (2, 4); 3) shared layers (9, 12, 15); 4) label

smoothing (0); 5) We set accumulate steps as 4. The best hyperparameters can be found in our code. For the learning rate scheduler, we use warm-up with linear decay. We set the number of warm-up steps as steps of an epoch. We run 50 epochs and choose the best epoch on the devset for testing. We use 4 random seeds: (0, 1, 2, 3) and report the average performances for each model. We finetune our model in a single Nvidia 3090 GPU with 24G memory or a single Nvidia A100 GPU with 40G memory (In our experiments, we did not find a clear difference in the performance of different GPUs). We modify our code based on Transfomers (Wolf et al., 2019) and Text2Event (Lu et al., 2021).