A Unified Multi-Task Learning Framework for Joint Extraction of Entities and Relations

Tianyang Zhao[1], Zhao Yan[2], Yunbo Cao[2], Zhoujun Li[1] *
[1]State Key Lab of Software Development Environment, Beihang University, Beijing, China
[2]Tencent Cloud Xiaowei, Beijing, China
{tyzhao,lizj}@buaa.edu.cn, {zhaoyan,yunbocao}@tencent.com

Abstract

Joint extraction of entities and relations has achieved great success in recent year by task decomposition and multi-task learning. Previous works effectively perform the task through different extraction order, such as relation-last, relation-first and relation-middle manner. However, these methods still suffer from the template-dependency, non-entity detection and non-predefined relation prediction problem. To overcome these challenges, in this paper, we propose a unified multi-task learning framework, which decomposes the task into three interacted sub-tasks. Specifically, we first introduce the type-attentional method for subject extraction to provide prior type information explicitly. Then, the subject-aware relation prediction is presented to select useful relations based on the combination of global and local semantics. Third, we propose a question generation based QA method for object extraction to obtain diverse queries automatically. Notably, our method detects subjects or objects without relying on NER models and thus it is capable of dealing with the non-entity scenario. Finally, three sub-tasks are integrated into a unified model through parameter sharing. Extensive experiments demonstrate that the proposed framework outperforms all the baseline methods on four benchmark datasets, and further achieves excellent performance for non-predefined relations.

Introduction

Entity-relation extraction aims to recognize entity spans from a sentence and detect the relations of entity pairs. Generally, it is formed as extracting an entity-relation triplet (e_1, r, e_2), which denotes that the relation r holds between the subject e_1 and the object e_2, i.e., (*Jack*, `Live-In`, *New York*). It has attracted increasing attention in recent years.

Early works mainly focus on pipelined methods, which divide the task into two independent sub-tasks, named entity recognition (NER) and relation extraction (RE) (Miwa et al. 2009; Chan and Roth 2011; Lin et al. 2016). However, the pipelined approaches neglect the inherent correlations between NER and RE tasks, which leads to the error propagation problem.

To leverage the interactions between NER and RE tasks, some recent works propose to extract entities and relations jointly. The joint methods tend to decompose the task into several fundamental procedures or solve the problem through multi-task learning framework. According to the extraction order of the triple elements, these models fall into three categories: relation-last, relation-first and relation-middle. The relation-last method can be formed as $(e_1, e_2) \rightarrow r$, which indentifies all entities in the sentence first using named entity recognition (NER) techniques, then conducts relation classification for any two entities (Katiyar and Cardie 2017; Eberts and Ulges 2019; Sun et al. 2019). However, the method requires to enumerate all pairs of entities, resulting in a heavy computational burden. And the redundant pairs cause negative influences on the relation classifier. The relation-first method is formed as $r \rightarrow (e_1, e_2)$. In this manner, the relation is generated initially by the Seq2Seq framework, and then subjects and objects are selected respectively based on the copy mechanism (Zeng et al. 2018; Zeng, Zhang, and Liu 2020). By predicting relations first, this kind of approaches filter out irrelevant relationships, which mitigate negative effects caused by useless relations and greatly avoid the data imbalance issue. Therefore, they have an overall higher computation efficiency. More recently, the relation-middle method, denoted as $e_1 \rightarrow r \rightarrow e_2$, has shown promising per-

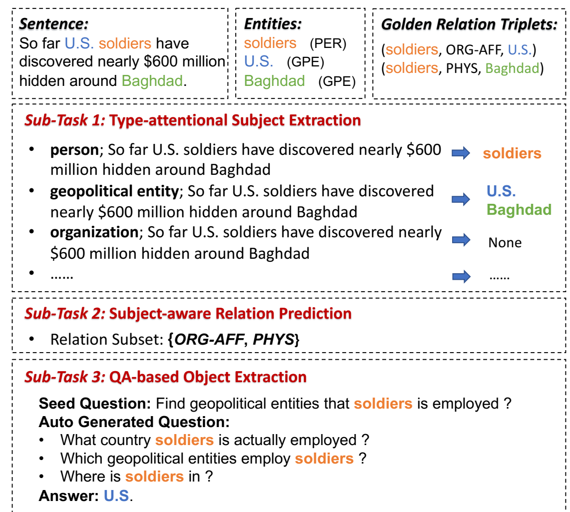

Figure 1: The extraction process based on the proposed multi-task learning framework

*Corresponding Author.

formance in relation extraction. Typical works extract the entity-relation triplet by multi-turn question answering (QA) structure (Levy et al. 2017; Li et al. 2019; Zhao et al. 2020). In this form, the subject and object are characterized by template-based queries. For example, by answering queries like `Which [person] is mentioned in the text?` and `Where was [person] born?` in turn, a triplet as (`[person]`, `born_in`, `[location]`) can be detected from the text. The QA-based structure exhibits advantages as follows: 1) The query question explicitly provides prior signals about the type information. 2) It enhances the interactions between the query and the text based on the QA structure. 3) It presents a natural way to deal with overlapping entities and relations(Zeng et al. 2018).

Despite the progresses made by these efforts, several major problems still remain to be solved. First, the QA-based approaches rely heavily on manually designed templates, making the model hard to transfer. Particularly, the study (Levy et al. 2017) proves that using diverse queries can improve model performance. In that case, designing multiple templates for every entity and relation type would largely increase the workforce. Second, present mainstreams have difficulties in identifying non-entity subjects or objects due to the heavy reliance of NER. In fact, relations can exist between any subject and object, including non-entity ones like text spans, time expressions, simple statements, and so on, especially in actual scenes. Third, existing systems mainly focus on predefined relations, so they can not deal with non-predefined relations that are not specified in advance. All mentioned above challenge the previous RE models.

To address the aforementioned problems, in this paper, we propose a comprehensive framework for joint entity and relation extraction. We follow the relation-middle based extraction order and decompose the task into three interrelated sub-tasks: the type-attentional subject extraction, the subject-aware relation prediction (SRP) and the QA-based object extraction. Figure 1 illustrates the process of our multi-task learning framework. Specifically, to alleviate the problem of relying on templates, we first present the type-attentional method to provide entity type information explicitly for the subject extraction task. Then, we propose a question generation (QG) strategy to obtain diverse queries automatically for the object extraction task. These two subtasks select text spans from the sentence based on the prior type information and query, instead of relying on NER. Hence, they also handle the non-entity problem effectively. Moreover, we design a distinctive way of fuzzy question answering to solve the non-predefined relation detection problem. Thirdly, we introduce the subject-aware relation prediction task to obtain a relation subset for the given subject using both global and local semantics. Finally, the three subtasks are integrated into a multi-task learning framework by sharing the feature encoder module. Empirical experiments show that the model better utilizes the inherent interactions among the sub-tasks and boosts the overall performance.

To summarize, the contributions of this work are:

- We define the entity-relation extraction into three interacted sub-tasks: the type-attentional subject extraction, the subject-aware relation prediction, and the QA-based object extraction, which effectively address the template-dependency, non-entity detection and non-predefined relation prediction issues.
- We present a multi-task learning framework to integrate the correlated sub-tasks and enhance the interactions among them.
- Extensive experiments show that our framework outperforms all the baseline models by a large margin. Detailed analyses further study the impact of extraction order for the relation extraction task.

Problem Definition

In this section, we define the problem formally. Denote $\mathcal{E}$ and $\mathcal{R}$ as the sets of predefined entity types and relation categories, respectively. For an input sentence $\boldsymbol{s} = \{s_1, s_2, \ldots, s_{N_s}\}$ with N_s tokens, the entity-relation extraction task aims to extract all relational triplets from the sentence. The relational triplet is formed as (e_i, r_{ij}, e_j), where $e_i \in \mathcal{E}, e_j \in \mathcal{E}, r_{ij} \in \mathcal{R}$, denoting that the relation r_{ij} holds between the subject e_i and the object e_j, e.g., the triplet (*Jack*, `Live-In`, *New York*). Especially, the entity pair (e_i, e_j) can associate with multiple relations.

Methodology

In this section, we elaborate on the structure of the proposed framework. To fully incorporate the inherent interactions among the related tasks, we employ the multi-task learning architecture to improve the overall performance. Based on the relation-middle extraction order, the framework consists of three interrelated tasks. 1) Type-attentional subject extraction, which provides type information explicitly to detect the subject entity from the sentence. 2) Subject-aware relation prediction, which is a multiple classification problem to select possible relations relevant to the given subject. 3) QA-based object extraction, which is a question answering problem to select the object entity from the sentence using auto-generated questions. Figure 2 shows an overview of the framework. In the following subsections, we first introduce the shared feature encoder. After that, we describe the structure of each task in detail.

Shared Feature Encoder

The shared feature encoder focuses on mapping input tokens to distributed semantic representations, consisting of a BERT layer and a contextual fusion layer. The learned feature is shared by the three downstream sub-tasks.

BERT Layer To better capture and generalize the semantics of the given sentence, we adopt BERT (Devlin et al. 2019) as the shared feature encoder, which is known as a pretrained language model based on bidirectional transformer structure and has achieved state-of-the-art performance on a wide range of NLP tasks. As illustrated in Figure 2, the input of BERT consists of three parts: the input sentence, the task-specific information and special tokens. Note that for the type-attentional subject extraction, the task-specific information is the entity type to be detected. While for the

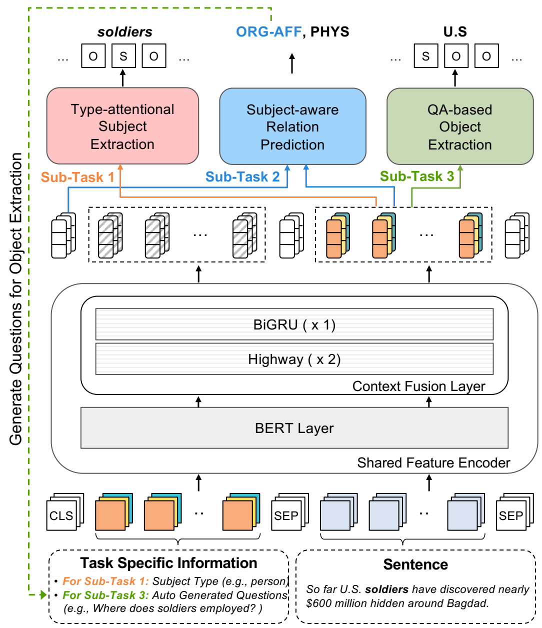

Figure 2: An overview of the multi-task learning framework. It consists of three interrelated tasks including the type-attentional subject extraction, the subject-aware relation prediction, and the QA-based object extraction. The three tasks are built on a shared feature encoder and have a task-specific output decoder, respectively. Here, we take the extraction procedure of the triplet (soldiers, ORG-AFF, U.S.) from ACE05 as an example.

QA-based object extraction, it becomes the auto-generated questions that query about the candidate object. Examples are shown in Figure 1. Formally, considering the sentence $\boldsymbol{s} = \{s_1, s_2, \ldots, s_{N_s}\}$, the input of BERT is the concatenation as:

$$\boldsymbol{x} = [\text{[CLS]}, \text{Task-specific Information}, \text{[SEP]}, \\ s_1, s_2, \ldots, s_{N_s}, \text{[SEP]}], \quad (1)$$

where [CLS] is the token will be used for subject-aware relation prediction, and [SEP] is a special separator. Then, the BERT yeilds the semantic representations for each token $\boldsymbol{h^b} = \{h_1^b, h_2^b, \ldots, h_{N_s}^b\}$ and the hidden state of the [CLS] token h_{CLS}, where $h_i \in \mathbb{R}^{d_h}$ with d_h as the dimension of the hidden state of BERT.

Context Fusion Layer To enhance the contextual information within the sequence efficiently, we apply a contextual fusion layer to further encode the output of BERT. The hidden states of BERT $\boldsymbol{h^b} = \{h_1^b, h_2^b, \ldots, h_{N_s}^b\}$ is firstly fed into a two-layer highway network (Srivastava, Greff, and Schmidhuber 2015):

$$\begin{aligned} \hat{h_i^b} &= H(h_i^b) * T(h_i^b) + h_i^b * (1 - T(h_i^b)) \\ T(h_i^b) &= \sigma(W^t h_i^b + b^t), \end{aligned} \quad (2)$$

where $W^t \in \mathbb{R}^{d_h \times d_t}$, $b^t \in \mathbb{R}^{d_t}$ are learned parameters with h_t as the hidden size of the highway network. Then, a one-layer bidirectional gate recurrent unit (BiGRU) (Cho et al. 2014) is followed by the highway network:

$$h_i = [\overrightarrow{\text{GRU}}(\hat{h_i^b}), \overleftarrow{\text{GRU}}(\hat{h_i^b})]. \quad (3)$$

The output h_i is the concatenation of the last hidden states for the forward GRU and the backward GRU. To this end, we obtain h_i as the final output of the shared feature encoder.

Type-attentional Subject Extraction

The type-attentional subject extraction task aims at detecting the subject entity from the sentence. Different from the previous works that need to design templated questions for each entity type, we simplify it into only using the text of entity type (e.g., person, location, organization, etc.). The most obvious advantages of the modification are: 1) without using any redundant words, it provides explicit entity type indicator for the self-attention structure; 2) it does not rely on any hand-crafted template and is both simple and effective. Specifically, the input of BERT is formed as:

$$\boldsymbol{x}_{\text{subject}} = [\text{[CLS]}, t, \text{[SEP]}, s_1, s_2, \ldots, s_{N_s}, \text{[SEP]}], \quad (4)$$

where $t \in \mathcal{E}$ is an entity type (i.e., person) with $\mathcal{E}$ as the set of entity types.

To detect the subject entity from the sentence, some previous works predict the starting and ending position of the sentence. However, this span-based method is limited to multi-answer scenario. Therefore, to tackle this problem, we predict a BIOES (Begin, Inside, Outside, Ending, Single) boundary for each token in the sentence. Formally, we employ a softmax layer to the contextual representations $\boldsymbol{h}$ and calculate the possibilities of all boundary tags for every token as:

$$\Pr(\text{tag} = y^\star | x_i) = \text{softmax}(W^n h_i + b^n), \quad (5)$$

where $W^n \in \mathbb{R}^{d_h \times d_n}$, $b^n \in \mathbb{R}^{d_n}$ with d_n as the number of boundary tags, i.e., $d_n = 5$. If all tags are predicted as O, then it means the sentence does not contain entities of the current type. Therefore, a set of subject entities with type t can be detected.

Subject-aware Relation Prediction

The subject-aware relation prediction (SRP) task focuses on predicting the relations that are relevant to the given subject, and thus the redundant relations can be filtered out. We predict the relations from both local and global features.

Local Relation Prediction (LRP) We first identify the probable relations for every given subject. For the subject e_i, the input to LRP is the concatenation of the local hidden represetation h_i and the learned entity type embedding $x_i^e \in \mathbb{R}^{d_e}$ with d_e as the dimention of x_i^e:

$$z_i = [h_i, x_i^e], \quad i = 1, \ldots, N_s, \quad (6)$$

where $z_i \in \mathbb{R}^{(d_h + d_e)}$. During training, we use the golden entity type. Then, the relations for e_i is calculated through a multi-sigmoid layer:

$$\Pr(\text{relation} = r_1, \ldots, r_k | e_i) = \sigma(W^l \cdot z_i + b^l), \quad (7)$$

where $W^l \in \mathbb{R}^{(d_h + d_e) \times |\mathcal{R}|}$, $b^l \in \mathbb{R}^{|\mathcal{R}|}$, $|\mathcal{R}|$ is the number of the relation types and $\sigma(\cdot)$ is the sigmoid funtion. Only the relation type with higher score than the threshold δ is kept as the candidate.

Global Relation Prediction (GRP) The previous work (Zhao et al. 2020) only uses the local information for relation prediction and neglects the semantic of the overall sentence. To address this problem, we introduce a global relation prediction to revise the learning of local relation classifier. Specifically, the hidden state corresponding to the first input token ([CLS]) can be considered as the aggregate representation of the whole sequence. Therefore, after obtaining the hidden output of [CLS] h_{CLS}, we predict the possible relations that is involved with the sentence $\boldsymbol{s}$ as:

$$\Pr(\text{relation} = \boldsymbol{r}|\boldsymbol{s}) = \sigma(W^g h_{\text{CLS}} + b^g), \quad (8)$$

where $\boldsymbol{r} \subseteq \mathcal{R}$, $W^g \in \mathbb{R}^{d_h \times |\mathcal{R}|}$, $b^r \in \mathbb{R}^{|R|}$.

Particularly, the local and global classifiers are integrated to calculate the loss during training. While for inference, we only use the relations obtained by the local predicter.

QA-based Object Extraction

After detecting the subject and possible relations, the QA-based object extraction task selects the object entity from the input sentence. In the following subsections, we introduce the procedure of QA-based object extraction and the automatic question generation in turn.

Object Extraction Process Formally, T questions $\boldsymbol{Q} = \{\boldsymbol{q_1}, \boldsymbol{q_2}, \ldots, \boldsymbol{q_T}\}$ are generated for object extraction. Each question q_t where $t \in 1, \ldots, T$ is concatencated with the input sentence $\boldsymbol{s}$ following Equation 1 as:

$$\boldsymbol{x}_{\text{object}} = [\text{[CLS]}, q_i, \text{[SEP]}, s_1, s_2, \ldots, s_{N_s}, \text{[SEP]}]. \quad (9)$$

Then, $\boldsymbol{x}_{\text{object}}$ is fed into the shared feature encoder. In the same way as Equation 5, the answer is obtained as $\boldsymbol{a_t} = \{a_{t1}, a_{t2}, \ldots, a_{tN_s}\}, t \in 1, \ldots, T$.

Additionally, to select the final answer from T answers, we employ a weighted voting strategy for answer selection following (Zhao et al. 2020). A weight w_t is defined for q_t, denoting the quality of the question. We calculate the F1 score f_t for q_t on the development set at the end of every training epoch. The weight w_t is updated as:

$$w_t = \sigma(f_t) * T, \quad (10)$$

where $\sigma(\cdot)$ is the sigmoid function. And the final answer $a_i^\star$ for position $i, i \in 1, \ldots, N_s$ is selected by weighted voting on the T tokens:

$$a_i^\star = \arg\max \sum_t w_t \cdot a_{ti}. \quad (11)$$

Therefore, the final answer is obtained as $\boldsymbol{a}^\star = \{a_1^\star, a_2^\star, \ldots, a_{N_s}^\star\}$.

Automatic Question Generation Different from the template-based methods, we propose to generate questions automatically based on a Seq2eq model. The advantages of the generation-based extraction manner are: 1) it saves a lot of manpower; 2) it can quickly generate any number of diverse questions to improve the model performance. In the following, we first introduce the structure of Seq2Seq model, and then describe the strategy for question generation.

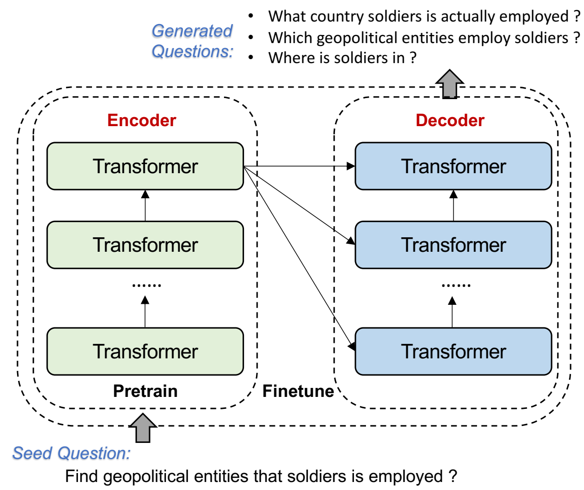

Figure 3: The architecture of the transformer-based Seq2Seq model for question generation. Here, we take the NL-question-driven generation as an example. The encoder takes a natural language question as input, and the decoder generates multiple paraphrased questions.

- **Transformer-based Seq2Seq.** As shown in Figure 3, we adopt a Seq2Seq structure with encoder and decoder both composed of Transformer layers (Rothe, Narayan, and Severyn 2019). The objective of the Seq2Seq is to predict multiple paraphrases based on the input seed question. For the encoder, we use the 12 transformer layers the same as BERT and initialize it with the pre-trained checkpoint. The decoder shares the same structure as the encoder but with all weights initialized randomly. The encoder and decoder use the identical embedding matrix initialized from the checkpoint. During training, we fine-tune the Seq2Seq on a question pairs task dataset, i.e., Quora [1].

- **Question Generation Strategy.** We propose two strategies for question generation. The first one is the **pseudo-question-driven generation**, which uses a simple pseudo-question as the seed question. The pseudo seed question is the combination of the subject text, the relation text, and the object type, such as `sodliers; organization affiliated; geographic political entity`. The strongest point of this formalization is that we can obtain any number of questions without human labor. However, considering the inner syntactic structure of questions, we further propose the **NL-question-driven generation**, which uses a natural language question as the seed question. It is obtained based on a general template as `Find [object type] that [subject text] is [relation type]`. Compared with the first one, this form can generate multiple grammatical questions, which may contain more syntactic information. Figure 3 shows an example of the NL-question-driven generation.

[1] https://www.kaggle.com/quora/question-pairs-dataset.

Data set	ALL Train/Test	|R|	|ET|	$\overline{\text{Triplet}}$
CoNLL04	910 / 288	5	4	1.37 / 1.41
ACE05	2600 / 583	6	7	1.81 / 1.92
DuIE2.0	171293 / 20674	55	17	1.88 / 1.90
Travel20	- / 200	8	1	- / 2.97

Table 1: Statistics of four datasets. Among them, R and ET denote the number of relation set and the number of entity type. $\overline{\text{Triplet}}$ is the average number of triplets that contained in each sentence.

Non-predefined Relation Detection

Specifically, to deal with non-predefined relations that are not observed beforehand, we propose a fuzzy question answering strategy. We first extract the subject through the aforementioned type-attentional approach. Then, we design default fuzzy questions to traverse the non-predefined relation and find the object in triplets. For example, the fuzzy question can be `What is the [relation type] of the [subject text]`? By filling the relation type and subject text, we finally extract the corresponding object through the QA-based method. In addition, we can customize the fuzzy questions of the non-predefined relations to make them more consistent with natural language questioning. Benefit from this strategy, we make obvious improvements in non-predefined relation detection.

Joint Training Objective

Overall, the input sentence is encoded by the shared BERT. The contextual outputs are used to calculate three tasks: type-attentional subject extraction, subject-aware relation prediction and QA-based Object Extraction. Therefore, we jointly optimize the integrated loss during training as:

$$\mathcal{L} = \mathcal{L}_{\text{Subject}} + \mathcal{L}_{\text{Global}} + \mathcal{L}_{\text{Local}} + \mathcal{L}_{\text{Object}}, \qquad (12)$$

where $\mathcal{L}_{\text{Subject}}$ denotes the final loss function. $\mathcal{L}_{\text{Global}}$ and $\mathcal{L}_{\text{Local}}$ denote the binary cross entropy for global and local relation classification, respectively. $\mathcal{L}_{\text{Object}}$ denotes the cross entropy loss for QA-based object extraction.

Experiment

In this section, we conduct extensive experiments to evaluate the effectiveness of the proposed multi-task framework.

Dataset

CoNLL04 (Roth and Yih 2004) and ACE05 (Doddington et al. 2004) are two widely used English benchmark datasets. We use the data split by (Gupta, Schütze, and Andrassy 2016) and (Miwa and Bansal 2016), respectively. DuIE2.0[2] and Travel20 are used for detailed analyses. DuIE2.0 is the largest Chinese RE dataset in the industry. Travel20 is an introductory data about attractions that includes only 200 texts and we use it for testing non-predefined relation extraction. The statistics of the datasets are listed in Table 1.

[2]https:/github.com/PaddlePaddle/Research/tree/master/KG/DuIE_Baseline.

Implemental Details

For all experiments, precision, recall, and micro-F1 score are adopted as our evaluation metrics for both entity and relation extraction. We initialize the BERT encoder layer using the pre-trained BERT-Base-Cased checkpoint [3] and therefore has 12 layers, a hidden size of 768. We use Adam optimizer with an initial learning rate of 5×10^{-5}. During training, we do warm-up startup first and employ a linearly decrease with 0.05 as the decay rate. For the model structure, we generate 5 questions for QA-based object extraction. The hidden dimension for highway and BiGRU are set as 400. The size of entity type embedding d_l is set as 50. The confidence threshold for relation classification δ is set as 0.3. For the question generation model, we adopt the BERT-Large-Cased checkpoint which has 24 layers, a hidden size of 1024. The training process is warm-started with $40K$ steps and is fine-tuned using Adam with a learning rate of 0.05.

Performance Comparison

Baselines We make comparisons with both non-QA-based methods and QA-based methods. Miwa et al. (2016) use the Tree-BiLSTM to learn context features using a dependency parser. Zhang et al. (2017) transfer the relation extraction task as a table-filling problem build a globally optimized model for end-to-end extraction. Models (Adel and Schütze 2017) utilizes the attentional LSTMs and normalized CNNs to capture features of entity pairs. Bekoulis et al. (2018) propose a multi-head selection framework to extract entities and relations simultaneously. Eberts and Ulges (2019) develop a span-based approach for joint entity-relation extraction based on BERT. (Sun et al. 2019) employs the graph convolutional network on an entity-relation bipartite graph structure. (Luan et al. 2019) adopts multi-task learning to identify coreferences, entities, relations using dynamic span graph. Li et al. (2019) cast the entity-relation extraction into a multi-turn question answering problem and tackle it with the QA-based method.

Main Results Table 2 shows the testing performance on both CoNLL04 and ACE05 datasets. As we can see, our unified multi-task framework outperforms all the baselines on CoNLL04 dataset, achieving the state-of-the-art performance. We also conduct a T-test to test statistical significance for the results, which shows the p-values are below the significance level $p < 0.05$, indicating the improvement is significant. Comparing with the QA-based model, **UMT w/ NLGQ** and **UMT w/ PseudoGQ** consistently surpass (Li et al. 2019) by 3.5% and 1.2% on the two datasets, showing that we obtain obvious improvement without using templated questions. Meanwhile, with the type-attentional subject extraction, we obtain strong competitive and even better performance than (Li et al. 2019). It reveals that the entity type is a useful indicator and can guide the self-attentive structure to detect the target tokens. To notice that, (Luan et al. 2019) uses external tools to ease the coreference problem. Differently, we still achieve considerable performance

[3]https://github.com/google-research/bert.

	Model	Entity			Relation		
		Precision	Recall	F1 score	Precision	Recall	F1 score
CoNLL04	(Adel and Schütze 2017)	-	-	82.1	-	-	62.5
	(Zhang, Zhang, and Fu 2017)	-	-	85.6	-	-	67.8
	(Bekoulis et al. 2018)	83.8	84.1	83.9	63.8	60.4	62.0
	(Eberts and Ulges 2019)	88.3	89.6	88.9	73.0	70.0	71.5
	(Li et al. 2019) w/ NLQ	89.0	86.6	87.8	69.2	68.2	68.9
	(Li et al. 2019) w/ PseudoQ	87.4	86.4	86.9	68.2	67.4	67.8
	UMT w/ NLGQ	88.7	88.8	88.8	72.9	71.6	**72.2**
	UMT w/ PseudoGQ	88.8	89.0	**88.9**	73.2	71.6	**72.4**
ACE05	(Miwa and Bansal 2016)	82.9	83.9	83.4	57.2	54.0	55.6
	(Zhang, Zhang, and Fu 2017)	-	-	83.5	-	-	57.5
	(Sun et al. 2019)	83.9	83.2	83.6	64.9	55.1	59.6
	(Luan et al. 2019)	-	-	88.4	-	-	**63.2**
	(Li et al. 2019) w/ NLQ	84.7	84.9	84.8	64.8	56.2	60.2
	(Li et al. 2019) w/ PseudoQ	83.6	84.7	84.2	60.4	55.9	58.1
	UMT w/ NLGQ	86.2	85.1	**85.7**	61.2	61.7	**61.4**
	UMT w/ PseudoGQ	86.2	85.1	**85.7**	61.0	61.3	**61.2**

Table 2: Performance comparisons on CoNLL04 and ACE05. UMT is the proposed method based on the multi-task learning framework. Here, w/ NLGQ denote models using questions generated by NL-question-driven strategy and w/ PseudoGQ denote models using questions generated by pseudo-question-driven strategy. w/ NLQ denotes models using one templated NL question and w/ PseudoQ denotes models using one pseudo question.

without using any NLP tools. All these above demonstrate the effectiveness of the proposed framework.

Ablation Study

In this subsection, we conduct ablation studies to discuss the effects of the multi-task combination and the impacts of several components. We adopt the **UMT w/ PseudoGQ** as the full model for comparison. Model with − *GRP* denotes the model dicard the global semantics and only use local information for relation classification. Model with − *GRP* − *LRP* denotes the model ablates the subject-aware relation prediction process and enumerates all relation types instead. Model with − *Context Fusion* denotes the model excluding the highway and BiGRU network. NER + SRP + MHE is a multi-task variant that replaces the QA-based object extraction with the multi-head selection (Bekoulis et al. 2018).

Model	Entity F1(Δ)	Relation F1(Δ)
CoNLL04		
UMT w/ PseudoGQ	**88.9**	**72.4**
− *GRP*	88.8 (-0.1)	72.2 (-0.2)
− *GRP* − *LRP*	88.7 (-0.2)	71.8 (-0.6)
− *Context Fusion*	87.9 (-1.0)	71.3 (-1.1)
NER + SRP + MHE	86.3 (-2.6)	64.7 (-7.5)
ACE05		
UMT w/ PseudoGQ	**85.7**	**61.2**
− *GRP*	85.6 (-0.1)	61.1 (-0.1)
− *GRP* − *LRP*	85.5 (-0.2)	60.3 (-0.9)
− *Context Fusion*	85.5 (-0.2)	60.7 (-0.5)
NER + SRP + MHE	84.5 (-1.2)	59.4 (-1.8)

Table 3: Ablation study results. GRP and LRP denote global relation prediction and loacal relation classification, respectively. SRP denotes subject-aware relation prediction which contains both GRP and LRP.

Table 3 shows the comparison results. We observe that, global relation prediction gives benefits to both datasets, proving that global information can guide the learning of local features. Meanwhile, without the subject-aware relation prediction, both entity and relation F1 drop significantly on two datasets. It demonstrates that filtering irrelevant relations is necessary for QA-based object extraction. Moreover, the context fusion layer (i.e., highway and BiGRU) contributes obviously by 0.9% and 0.5% to the performance, revealing that the combination of highway and BiGRU can further encode the features of BERT output. Additionally, our multi-task combination is much superior to NER + SRP + MHE. It further explains the rationality of our framework.

Analyses on Extraction Order and Non-predefined Relations

Impact of Extraction Order Note that, the proposed UMT follows the relation-middle extraction order. Therefore, we take the relation-first model (UMT$_{RF}$) for comparison and conduct experiments on CONLL04, ACE05 and DuIE2.0. Specifically, UMT$_{RF}$ conducts relation prediction initially to narrow the relation set. Then, it follows by the same subject extractor and the pseudo generated question based object extractor. Table 4 shows the experimental result. For clarity, we rename the UMT w/ PseudoGQ as UMT$_{RM}$ to highlight its relation-middle extraction manner. We can observe that, the UMT$_{RM}$ outperforms the UMT$_{RF}$ on CONLL04 and ACE05 by a large margin. While UMT$_{RF}$ beats UMT w/ PseudoGQ obviously on DuIE2.0. The results demonstrate that the relation-middle method (UMT) is more qualified for concise datasets with fewer entity and relation types. Differently, the number of relation type on DuIE2.0 is up to 55. Such a large relation set greatly increases the computational burden of UMT$_{RM}$ and makes it difficult to achieve good results. Hence, for much complicated datasets,

Dataset	UMT$_{RM}$ P / R / F1	UMT$_{RF}$ P / R / F1
CoNLL04	**73.2** / 71.6 / **72.4**	63.4 / **73.4** / 68.0
ACE05	**61.0** / 61.3 / **62.1**	59.5 / **61.9** / 60.7
DuIE2.0	66.9 / 63.0 / 64.9	**76.1** / **74.9** / **75.5**

Table 4: Comparisons of relation-middle model against relation-first model. For clarity, we rename UMT w/ PseudoGQ as UMT$_{RM}$.

Action	Precision	Recall	F1-score
BERT	47.82	49.61	48.7
UMT w/ PseudoGQ	85.5	75.4	80.1
UMT w/ NLGQ	90.3	86.2	88.2

Table 5: Performane comparisons of non-predefined relation extraction on Travel20 dataset.

predicting relation firstly may lead to better performance.

Performance on Non-predefined Relations To evaluate the performance of UMT on non-predefined relations, we test the model trained with ACE05 on a new Travel20 dataset, whose relation types are totally different from ACE05. As shown in Table 5, UMT still achieves considerable results on the new dataset. Meanwhile, UMT w/ NLGQ gains better performance that UMT w/ PseudoGQ. While existing methods fail to solve such a problem. The results demonstrate that our UMT also works for the non-predefined relations, and the syntactic information of natural language helps to further improvements.

Impact of the Number of Questions

In this subsection, we discuss the impact of the number of generated questions. We perform evaluations by varying the question numbers T as 1, 3, 5, respectively. The results of the relation F1 are shown in Figure 4. Notably, asking diverse questions obviously improves the model performance on the two datasets. By using 5 questions, we can further gain a performance boost. These well verify the effectiveness of diverse question answering mechanism. Note that, the ACE05 is more sensitive to NL generated questions. We consider that the relation types in ACE05 are more confusing, as one relation can further be divided into several subtypes. Therefore, the NL questions can provide structural information to help the semantic understanding.

Related Work

Extracting entities and their relations from the unstructured text is an essential task for natural language understanding. Early pipelined works suffer from error propagation problems. Recently, many joint models have been proposed. A majority of joint learning strategies have been well studied, such as parameter-sharing strategy (Miwa and Bansal 2016; Katiyar and Cardie 2017), joint decoding algorithms (Yang and Cardie 2013; Katiyar and Cardie 2016), and global normalization (Zhang, Zhang, and Fu 2017; Ren et al. 2017).

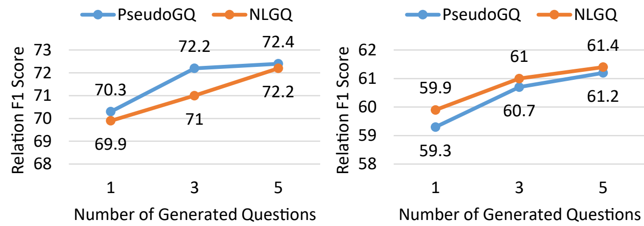

Figure 4: Relation F1 for the number of questions $T = 1, 3, 5$ on CONLL04 (left figure) and ACE05 (right figure).

Most works detect entities by sequence labeling approaches. Methods for relation extraction can be fell into two categories. The first category treats the relation extraction as a classification problem. For example, Miwa et al. (2016) adopt the Tree-BiLSTM based on a dependency analysis. Eberts and Ulges (2019) propose a span-based method with pre-trained transformers. Another category cast the relation extraction as a question asking problem. Levy et al. (2017) firstly propose to reduce the relation extraction to answering simple machine reading comprehension questions. Then, Li et al. (2019) introduce a multi-turn question answering framework for entity-relation extraction. More recently, Zhao et al. (2020) further improve it by asking diverse questions and achieving a significant performance boost.

Our work is different from previous works (Zhao et al. 2020) and enjoys the following new keypoints. First, we propose the type-attentional subject extraction and generation based object extraction without relying on the hand-crafted templates. Second, they only use the local context feature for relation classification, while we further introduce the global information to guide the feature learning process. Finally, we divide the entity-relation extraction into three sub-tasks and integrate them into a multi-learning framework, which can better capture the correlations among the sub-tasks.

Conclusion

This paper proposes a unified multi-task learning framework for entity- relation extraction. we first introduce a type-attentional subject extraction task for subject detection. Then, we present a subject-aware relation prediction task to filter out irrelevant relations for subject entities. After that, we propose a question generation based method for object extraction. Finally, these interacted tasks are integrated into a unified multi-task learning framework. Extensive experiments on benchmark datasets verify the effectiveness the proposed framework.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant Nos.U1636211, 61672081, 61370126), the Beijing Advanced Innovation Center for Imaging Technology (Grant No.BAICIT2016001), and the Fund of the State Key Laboratory of Software Development Environment (Grant No.SKLSDE2019ZX-17). We thank anonymous reviewers for their helpful comments.

References

Adel, H.; and Schütze, H. 2017. Global Normalization of Convolutional Neural Networks for Joint Entity and Relation Classification. In *EMNLP*, 1723–1729.

Bekoulis, G.; Deleu, J.; Demeester, T.; and Develder, C. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications* 114: 34–45.

Chan, Y. S.; and Roth, D. 2011. Exploiting syntactico-semantic structures for relation extraction. In *ACL*, 551–560.

Cho, K.; Van Merriënboer, B.; Bahdanau, D.; and Bengio, Y. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* .

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 4171–4186.

Doddington, G. R.; Mitchell, A.; Przybocki, M. A.; Ramshaw, L. A.; Strassel, S. M.; and Weischedel, R. M. 2004. The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation. In *Lrec*.

Eberts, M.; and Ulges, A. 2019. Span-based Joint Entity and Relation Extraction with Transformer Pre-training. *arXiv preprint arXiv:1909.07755* .

Gupta, P.; Schütze, H.; and Andrassy, B. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *COLING*, 2537–2547.

Katiyar, A.; and Cardie, C. 2016. Investigating LSTMs for joint extraction of opinion entities and relations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 919–929.

Katiyar, A.; and Cardie, C. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *ACL*, 917–928.

Levy, O.; Seo, M.; Choi, E.; and Zettlemoyer, L. 2017. Zero-Shot Relation Extraction via Reading Comprehension. In *CoNLL*, 333–342.

Li, X.; Yin, F.; Sun, Z.; Li, X.; Yuan, A.; Chai, D.; Zhou, M.; and Li, J. 2019. Entity-Relation Extraction as Multi-Turn Question Answering. *arXiv preprint arXiv:1905.05529* .

Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; and Sun, M. 2016. Neural relation extraction with selective attention over instances. In *ACL*, 2124–2133.

Luan, Y.; Wadden, D.; He, L.; Shah, A.; Ostendorf, M.; and Hajishirzi, H. 2019. A general framework for information extraction using dynamic span graphs. In *NAACL*, 3036–3046.

Miwa, M.; and Bansal, M. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. *arXiv preprint arXiv:1601.00770* .

Miwa, M.; Sætre, R.; Miyao, Y.; and Tsujii, J. 2009. A rich feature vector for protein-protein interaction extraction from multiple corpora. In *EMNLP*, 121–130.

Ren, X.; Wu, Z.; He, W.; Qu, M.; Voss, C. R.; Ji, H.; Abdelzaher, T. F.; and Han, J. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th International Conference on World Wide Web*, 1015–1024.

Roth, D.; and Yih, W.-t. 2004. A Linear Programming Formulation for Global Inference in Natural Language Tasks. In *CoNLL*, 1–8.

Rothe, S.; Narayan, S.; and Severyn, A. 2019. Leveraging pre-trained checkpoints for sequence generation tasks. *arXiv preprint arXiv:1907.12461* .

Srivastava, R. K.; Greff, K.; and Schmidhuber, J. 2015. Highway networks. *arXiv preprint arXiv:1505.00387* .

Sun, C.; Gong, Y.; Wu, Y.; Gong, M.; Jiang, D.; Lan, M.; Sun, S.; and Duan, N. 2019. Joint type inference on entities and relations via graph convolutional networks. In *ACL*, 1361–1370.

Yang, B.; and Cardie, C. 2013. Joint inference for fine-grained opinion extraction. In *ACL*, 1640–1649.

Zeng, D.; Zhang, H.; and Liu, Q. 2020. CopyMTL: Copy Mechanism for Joint Extraction of Entities and Relations with Multi-Task Learning. In *AAAI*.

Zeng, X.; Zeng, D.; He, S.; Liu, K.; and Zhao, J. 2018. Extracting Relational Facts by an End-to-End Neural Model with Copy Mechanism. In *ACL*, 506–514.

Zhang, M.; Zhang, Y.; and Fu, G. 2017. End-to-end neural relation extraction with global optimization. In *EMNLP*, 1730–1740.

Zhao, T.; Yan, Z.; Cao, Y.; and Li, Z. 2020. Asking Effective and Diverse Questions: A Machine Reading Comprehension based Framework for Joint Entity-Relation Extraction. In *IJCAI*.