TERL: Transformer Enhanced Reinforcement Learning for Relation Extraction

Yashen Wang
China Academy of Electronics and Information Technology of CETC, Artificial Intelligence Institute of CETC
yswang.arthur@gmail.com

Tuo Shi
Beijing Police College
shituo@bjpc.edu.cn

Xiaoye Ouyang
National Engineering Laboratory for Risk Perception and Prevention (RPP), China Academy of Electronics and Information Technology
ouyangxiaoye@cetc.com.cn

Dayu Guo *
CETC Academy of Electronics and Information Technology Group Co.,Ltd.
guodayu1@cetc.com.cn

Abstract

Relation Extraction (RE) task aims to discover the semantic relation that holds between two entities and contributes to many applications such as knowledge graph construction and completion. Reinforcement Learning (RL) has been widely used for RE task and achieved SOTA results, which are mainly designed with rewards to choose the optimal actions during the training procedure, to improve RE's performance, especially for low-resource conditions. Recent work has shown that offline or online RL can be flexibly formulated as a sequence understanding problem and solved via approaches similar to large-scale pre-training language modeling. To strengthen the ability for understanding the semantic signals interactions among the given text sequence, this paper leverages Transformer architecture for RL-based RE methods, and proposes a generic framework called **T**ransformer **E**nhanced **RL** (**TERL**) towards RE task. Unlike prior RL-based RE approaches that usually fit value functions or compute policy gradients, TERL only outputs the best actions by utilizing a masked Transformer. Experimental results show that the proposed TERL framework can improve many state-of-the-art RL-based RE methods.

1 Introduction

Relation Extraction (RE) aims to discover the binary semantic relation between two entities in a sequence of words. E.g., given a sentence "$\cdots$ Carey will succeed Cathleen P. Black, who held the position for 15 years and will take on a new role as chairwoman of Hearst Magazines, the company said$\cdots$" (Xue et al., 2020), and we aim to predict the relation type between two entities "Cathleen P. Black" and "chairwoman" and the result is "per:title".

Deep neural network (DNN) driven methods have gained decent performance when labeled data is available (Hu et al., 2021b; Guo et al., 2020). While Reinforcement Learning (RL) based RE methods gain a lot of attention recently and show encouraging effects (Takanobu et al., 2018; Hu et al., 2021b; Wang and Zhang, 2021), especially in low-resource and few-shot conditions. Since this kinds of work requires *fewer* labeled data or could expand limited labeled data by exploiting information on unlabeled data to iteratively improve the performance (Hu et al., 2021b).

Recent works have shown Transformers (Vaswani et al., 2017) can model high-dimensional distributions of semantic concepts at scale, and several attempts have demonstrated the combination between transformers and RL architecture (Parisotto and Salakhutdinov, 2021; Parisotto et al., 2020; Zambaldi et al., 2019). These works have shown that the Transformer's efficiency for modeling beneficial semantic interactions in the given sequence (Chen et al., 2021a; Zheng et al., 2022), which is very enlightening for RE task. Given the diversity of successful applications of such models (Chen et al., 2021a), this paper seeks to investigate their application to sequential RE problems formalized as RL, because of the three main advantages of transformers: (i) Its ability to model long sequences has been demonstrated in many tasks; (ii) It could perform long-term *credit assignment* via self-attention strategy, contrary to Bellman backups (Lee et al., 2021) which slowly propagate rewards and are prone to distractor signals (Hung et al., 2019) in Q-learning, which could enable Transformer-based architecture to still work effectively in the presence of distracting rewards (Chen et al., 2021a); and (iii) It can model a wide distribution of behaviors, enabling better generalization (Ramesh et al., 2021). Hence, inspired by (Chen et al., 2021a;

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 677-688, Harbin, China, August 3 - 5, 2023.

Zheng et al., 2022), we try to view the RL-based RE as a conditional sequence understanding problem. Especially, we model the joint distribution of the sequence of states, actions and rewards, and discuss whether generative sequence understanding could can serve as a substitute for traditional RL algorithms in RE task. Overall, we propose **T**ransformer **E**nhanced **R**einforcement **L**earning (**TERL**), which abstracts RL paradigm as autoregressively sequence understanding and utilize Transformer architecture in BERT [1] to model text sequences with minimal modification to native transformer's architecture, and we investigate whether the sequence understanding paradigm can perform policy optimization by evaluating TERL on RL benchmarks in RE task. This enables us to leverage the scalability of the Transformer's architecture, as well as the related advancements in pre-training language modeling (such as the BERT's series).

Especially, following the backbone proposed in (Chen et al., 2021a), we train Transformer architecture on collected experience with a sequence understanding objective for RE task, instead of training a policy through conventional RL algorithms (Hu et al., 2021b; Wang and Zhang, 2021). This transformer is trained to predict next token in a sequence of rewards (forward-cumulative-rewards emphasized here), states, and actions. This paper shows that leveraging Transformers can open up another paradigm to solve RL-based RE problem. The main differences between this work and previous RL-based RE methods, can be concluded as follows: (i) RL is transformed into sequence understanding; (ii) We learn the natural projection from reward and state to action, instead of maximizing cumulative discount rewards or *only* modeling state and action in conventional behavior cloning paradigm (Chen et al., 2021b); (iii) Q/V-functions are *no* need to be learned, while we directly model it as a sequence problem, wherein as long as given the expected return, we can get the corresponding action; and (iv) Bellman backups or other temporal difference frameworks is *no* need; In RE tasks (even relation and entity joint extraction tasks) with our work, the expected target return is highly correlated with the actual observed return. Under certain conditions, the proposed TERL could successfully generate sequences that almost completely match the required returns. In addition, we can prompt TERL with a higher return than the maximum event available in the dataset, indicating that our TERL can sometimes be extrapolated. Moreover, the proposed framework can also be used as a plug-in unit for any RL-based RE architecture, and be extended to relation and entity joint extraction task (Zhou et al., 2019). Experimental results show that the proposed TERL framework can improve many state-of-the-art RL-based RE methods.

2 Related Work

Relation Extraction (RE) aims to predict the binary relation between two entities in a sequence of words. Recent work leverages deep neural network (DNN) for learning the features among two entities from sentences, and then classify these features into pre-defined relation types (Hu et al., 2021b). These methods have achieved satisfactory performance when labeled data is sufficient (Zeng et al., 2015; Guo et al., 2020), however, it's labor-intensive to obtain large amounts of manual annotations on corpus. Hence, few-shot (even zero-shot) RE methods gained a lot of attention recently, since these methods require *fewer* labeled data and could expand limited labeled information by exploiting information on unlabeled data to iteratively improve the performance. Wherein, Reinforcement Learning (RL) based methods have grown rapidly (Zeng et al., 2019; Wang and Zhang, 2021), which has been widely used in Nature Language Processing (NLP) (Narasimhan et al., 2016; Zhou et al., 2019; Li et al., 2021). These methods are all designed with rewards to force the correct actions to be chosen during the model's training procedure. For RE task, (Qin et al., 2018) proposes a RL strategy to generate the false-positive indicator, where it automatically recognizes false positives for each relation type without any supervised information. (Li et al., 2021) addresses the RE task by capturing rich contextual dependencies based on the attention mechanism, and using distributional RL to generate optimal relation information representation. (Hu et al., 2021b) proposes gradient imitation RL method to encourage pseudo label data to imitate the gradient descent direction on labeled data.For relation and entity joint extraction task, (Takanobu et al., 2018) proposes a hierarchical RL framework which decomposes the whole extraction process into a hierarchy of two-level RL policies for relation extraction and entity extraction, respectively. (Zeng et al., 2019) applies policy gradient method to model future reward in a joint entity and relation extraction task. (Wang

[1] Other transformer architecture is also applicable.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 677-688, Harbin, China, August 3 - 5, 2023.

and Zhang, 2021) jointly extracts entities and relations, and propose a novel bidirectional interaction RL model.

Recently, there exist many exciting works which formulate the Reinforcement Learning (RL) problem as a context-conditioned "sequence understanding" problem (Chen et al., 2021a; Zheng et al., 2022). For *offline* RL settings, (Chen et al., 2021a) trains a transformer (Vaswani et al., 2017) as a model-free context-conditioned policy, and (Janner et al., 2021) trains a transformer as both a policy and model and shows that beam search can be used to improve upon purely model-free performance. These works focus on exploring *fixed* datasets that transformers are traditionally trained with in NLP applications, which is similar to our focus. For *online* RL settings, (Zheng et al., 2022) proposes a RL algorithm based on sequence understanding that blends offline pre-training with online fine-tuning in a unified framework. To best of our knowledge, this work is the *first* test to leverage Transformer for enhancing RL-based RE task.

3 Methodology

This section presents the proposed TERL for RE task, as summarized in Fig. 1.

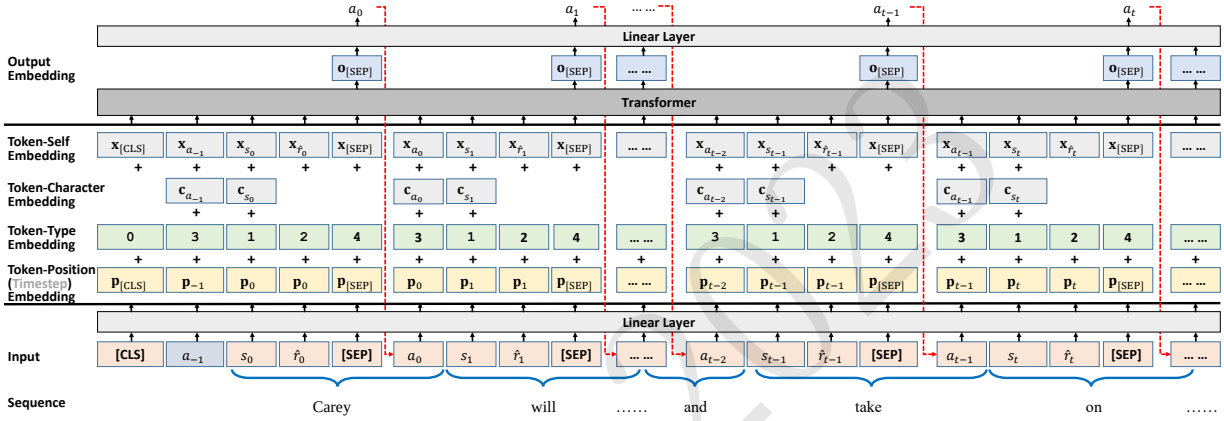

Figure 1: The architecture of TERL for RE task.

3.1 Relation Extraction with RL

The RL policy π for Relation Extraction (RE), usually aims to detect the relations in the given word sequence $\tau_1 = \{w_0, w_1, w_2, \cdots, w_T\}$, which can be regarded as a conventional RL policy over actions. A Markov Decision Process (MDP) described by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$ (Wang and Zhang, 2021), is usually used for learning procedure. Especially, the MDP tuple consists of states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, transition probability $P(s'|s, a)$ and rewards $r \in \mathcal{R}$. At timestep t, s_t, a_t, and $r_t = \mathcal{R}(s_t, a_t)$ denote the state, action, and reward, respectively. The goal in RL is to learn a desired policy which maximizes the expected reward $\mathbb{E}(\sum_{t=1}^{T} r_i)$ in MDP (Chen et al., 2021a).

Action: The action a_t is selected from $\mathcal{A} = R \bigcup \texttt{None}$, wherein notation `None` indicates that *no* relation exists in the given context, and R is the pre-defined relation-type set.

State: The state $s_t \in \mathcal{S}$ of the relation extraction RL process at timestep t, can be represented by (Wang and Zhang, 2021; Takanobu et al., 2019): (i) the current hidden state vector $\mathbf{h}_t$, (ii) the relation-type vector $\mathbf{a}_{t-1}$ (the embedding of the latest action a_{t-1} that $a_{t-1} \neq \texttt{None}$, a learnable parameter), and (iii) the state from the last timestep $\mathbf{s}_{t-1}$, formally represented as follows:

$$\mathbf{s}_t = f(\mathbf{W}_{\mathcal{S}}[\mathbf{h}_t; \mathbf{a}_{t-1}; \mathbf{s}_{t-1}]) \tag{1}$$

where $f(\cdot)$ denotes a non-linear function implemented by MLP (Other encoder architecture is also applicable, which is not the focus of this paper). To obtain the current hidden state $\mathbf{h}_t$, sequence Bi-LSTM over the current input word embedding $\mathbf{x}_t$, character embedding $\mathbf{c}_t$, token-type embedding $\mathbf{v}_t$, and

token-position embedding $\mathbf{p}_t$, can be used here, as follows:

$$\begin{aligned}\overrightarrow{\mathbf{h}_t} &= \overrightarrow{\texttt{LSTM}}(\overrightarrow{\mathbf{h}_{t-1}}, \mathbf{x}_t, \mathbf{c}_t, \mathbf{v}_t, \mathbf{p}_t) \\ \overleftarrow{\mathbf{h}_t} &= \overleftarrow{\texttt{LSTM}}(\overleftarrow{\mathbf{h}_{t+1}}, \mathbf{x}_t, \mathbf{c}_t, \mathbf{v}_t, \mathbf{p}_t) \\ \mathbf{h}_t &= [\overrightarrow{\mathbf{h}_t}; \overleftarrow{\mathbf{h}_t}]\end{aligned} \tag{2}$$

Policy: The stochastic policy for detecting relation-type can be defined as $\pi : \mathcal{S} \rightarrow \mathcal{A}$, which specifies a probability distribution over actions:

$$a_t \sim \pi(a_t|s_t) = \text{SoftMax}(\mathbf{W}_\pi \mathbf{s}_t) \tag{3}$$

Reward: The environment provides intermediate reward r_t to estimate the future return when chose action a_t. The reward is computed as follows:

$$r_t = \begin{cases} 1, & a_t \text{ conforms to } \tau_1, \\ 0, & a_t = \texttt{None}, \\ \text{-1}, & a_t \text{ not conforms to } \tau_1. \end{cases} \tag{4}$$

If a_t equals to `None` at certain timestep t, the agent transfers to a new relation extraction state at the next timestep $t+1$. Such a MDP procedure mentioned above continues until the *last* action about the *last* word w_T of current sequence is sampled. Finally, a final reward r_* is obtained to measure the RE's performance that the RL's policy π detects, which is obtained by the weighted harmonic mean of precision and recall in terms of the relations in given sentence sequence τ_1 (Wang and Zhang, 2021): $r_* = \frac{(1+\beta^2)\cdot Prec \cdot Rec}{\beta^2 \cdot Prec + Rec}$. Wherein, notation $Prec$ and Rec indicate the precision value and recall value respectively, computed over the current sequence τ_1.

3.2 Transformer

For simplicity, we take BERT as an example. BERT (Devlin et al., 2019) is the first bidirectional language model, which makes use of left and right word contexts simultaneously to predict word tokens. It is trained by optimizing Masked Language Model (MLM) objective etc.,. The the architecture of conventional BERT is a multi-layer bidirectional transformer encoder (Vaswani et al., 2017), and the inputs are a sequence of tokens $\{x_1, x_2, \cdots, x_n\}$. The tokens go through several layers of *transformers*. At each layer, a new contextualized embedding is generated for each token by weighted-summing all other tokens' embeddings. The weights are decided by several attention matrices (*multi-head attention*). Note that: (i) tokens with *stronger* attentions are considered *more* related to the target word; (ii) *Different* attention matrices capture *different* types of token relations, such as exact match and synonyms.

The entire BERT model is pre-trained on large scale text corpora and learns linguistic patterns in language. It can be viewed as an interaction-based neural ranking model (Guo et al., 2016). Given the widespread usage of BERT, we do not detail the architecture here. See (Devlin et al., 2019) for more details about the conventional architecture of BERT and its variants for various applications.

3.3 Input Generation

Given sequence under RL's paradigm $\{s_0, r_0, a_0, s_1, r_1, a_1, \cdots, s_T, r_T, a_T\}$, the reward of a sequence at step t, is defined as the forward-cumulative-rewards from the current timestep, similar to (Chen et al., 2021a): $\hat{r}_t = \sum_{i=t}^{T} r_i$, without discount. Wherein, r_i denotes the reward from environment at timestep i. Because we want to generate actions based on *future* (forward direction) expected returns rather than *past* (backward direction) rewards. Hence the input sequence towards our Transformer, is defined as follows, which consists of states, actions and rewards:

$$\tau = \{a_{-1}, s_0, \hat{r}_0, a_0, s_1, \hat{r}_1, a_1, s_2, \hat{r}_2, a_2, \cdots, s_T, \hat{r}_T, a_T\} \tag{5}$$

It represents the whole sequence from the beginning to the end, but in the actual training process, we often only intercept K timesteps (i.e., context length) as input (details in Sec. 3.4). Wherein, K is a hyper-parameter with different values towards different tasks, and a_{-1} in E.q. (5) is a padding indicator.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 677-688, Harbin, China, August 3 - 5, 2023.

3.4 Procedure

We feed the last K timesteps into TERL, for a total of $3 \times K$ tokens (one for each type: states, actions and rewards). As shown in Fig. 1, to obtain token embeddings: (i) for state and action, E.q. (1) and E.q. (2) are used to generate state embedding and action embedding, which consider word embeddings, character embeddings, type embeddings and position embeddings (Wang and Zhang, 2021; Zhou et al., 2019); (ii) for forward-cumulative-rewards, we learn a linear-layer, which projects inputs to the embedding dimension, followed by layer-normalization (Chen et al., 2021a; Xiong et al., 2020).

Moreover, a token-position (respect to timestep) embedding, a token-type embedding for each token as well as a token-character embedding respect to action token or state token, is learned and added to each token, as one timestep corresponds to 4 types of tokens in our framework. Wherein, we define the token-type projection as: $\{[\mathrm{CLS}], action, state, reward, [\mathrm{SEP}]\} \rightarrow \{0, 1, 2, 3, 4\}$. The tokens are then processed by a BERT (Devlin et al., 2019) or GPT (Radford and Narasimhan, 2018) model (as well as their variants), which predicts future (forward) action: $\{a_{t-1}, s_t, \hat{r}_t\} \rightarrow a_t$.

With efforts above, after executing the generated actions for the current state, we reduce the target return by the rewards we receive and repeat until the end of the episode. The output is action sequence $\{a_0, a_1, a_2, \cdots, a_T\}$, which is generated with a linear layer (on top of Fig. 1). Note that, the output can also includes sequence of states or rewards. For simplicity, we do not use them and leave for future discussion.

The details about training procedure and testing procedure, can be concluded as follows:

(i) In training procedure, we sample mini-batches of sequence length K (i.e., context length) from the training dataset, and mainly use the self-attention paradigm in Transformer. a_{-1} with zero-padding is added before the entire sequence. As shown in Fig. 1, predicting action at each timesetp a_t with cross-entropy loss, is used as the training objective.

(ii) At test time, we use the definition of E.q. (4) as the desired performance. At the beginning, given the desired performance (e.g., $\hat{r}_0 = 1$) as well as the initial state s_0, transformer generates action a_0. Let the agent perform actions a_0, the environment will give return r_0 and the next state s_0, and we can get $\hat{r}_1$. Then $\{a_0, s_1, \hat{r}_1\}$ can be added into the input sequence, and we can get a_1. The aforementioned testing procedure is *autoregressive*, because the output a_{t-1} in previous timestep will intuitively be the viewed as input in the following timestep: $\{a_{t-1}, s_t, \hat{r}_t\} \rightarrow a_t$.

4 Experiments

This paper constructs relation extraction task and relation and entity joint extraction task for evaluations.

4.1 Datasets and Metrics

For relation extraction (RE) task examination, we follow (Hu et al., 2021b) to leverage two public RE datasets for conducting experiments on, including SemEval 2010 Task 8 (SemEval) (Hendrickx et al., 2010), and TAC Relation Extraction Dataset (TACRED) (Zhang et al., 2017): (i) SemEval dataset is a standard benchmark dataset for testing RE models, which consists of training, validation and test set with 7,199, 800, 1,864 relation-mentions respectively, with totally 19 relations types (including `None`). (ii) TACRED dataset is a more large-scale crowd-sourced RE dataset, which is collected from all the prior TAC KBP relation schema. It consists of training, validation and test set with 75,049, 25,763, 18,659 relation-mentions respectively, with totally 42 relation types (including `None`).

We also test the extension of the proposed framework for relation and entity joint extraction task. For this task, we conduct experiments on two public datasets NYT (Riedel et al., 2010) and WebNLG (Gardent et al., 2017): (i) NYT dataset is originally produced by a distant supervision method, which consists of 1.18M sentences with 24 predefined relation types; (ii) WebNLG dataset is created by Natural Language Generation (NLG) tasks and adapted by (Zeng et al., 2018) for relational triple extraction task. It consists of 246 predefined relation types.

For both datasets, we follow the evaluation setting used in previous works. A triple (head entity, relation-type, tail entity) is regarded as *correct* if the relation-type (belongs to R) and the two corresponding entities (head entity and tail entity head entity) are all correct. Precision, Recall and F1-score are introduced here

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 677-688, Harbin, China, August 3 - 5, 2023.

as metrics for all the compared models. For each dataset, we randomly chose 0.5% data from the training set for validation (Wang and Zhang, 2021).

4.2 Baselines

For relation extraction task, the baselines include three categories:

(i) When comparing with supervised relation encoders with only labeled data, we choose **LSTM**(Hochreiter and Schmidhuber, 1997), **PCNN**(Zeng et al., 2015), **PRNN**(Zhang et al., 2017), and **BERT**(Devlin et al., 2019) as baselines.

(ii) When comparing with semi-supervised relation encoders with both labeled data and unlabeled (or pseudo labeled) data, we choose **Self-Training**(Rosenberg et al., 2005), **Mean-Teacher**(Tarvainen and Valpola, 2017), **DualRE**(Lin et al., 2019), and **MetaSRE**(Hu et al., 2021a) as baselines.

(iv) When comparing with the RL-based models, we choose **RDSRE**(Qin et al., 2018), **DAGCN**(Li et al., 2021) and **GradLRE**(Hu et al., 2021b) as baselines. **RDSRE** is a RL strategy to generate the false-positive indicator, where it automatically recognizes false positives for each relation type without any supervised information. **DAGCN** addresses the RE task by capturing rich contextual dependencies based on the attention mechanism, and using distributional RL to generate optimal relation information representation. **GradLRE** is gradient imitation RL method to encourage pseudo-label data to imitate the gradient descent direction on labeled data and bootstrap its optimization capability through trial and error. As our work can be viewed as a plug-in unit for this kind of RL-based model, the variant model with help of our work is named with suffix "**+TERL**".

Our framework can be easily extended to relation and entity joint extraction method based on RL. For evaluating joint extraction task, the baselines include four categories:

(i) The traditional pipeline models are **FCM** (Kim, 2014) and **LINE** (Gormley et al., 2015). Wherein, **FCM** is a conventional and compositional joint model by combining hand-crafted features with learned word embedding for relation extraction task. **LINE** is a network embedding method which embeds very large information networks into low-dimensional vectors. Note that, following (Wang and Zhang, 2021), both of them obtain the NER results by CoType (Ren et al., 2017), and then the results are fed into the two models to predict relations.

(ii) The joint learning baselines used here include feature-based methods (e.g., **DS-Joint** (Yu and Lam, 2010), **MultiR** (Hoffmann et al., 2011) and **CoType** (Ren et al., 2017)), and neural-based methods (e.g., **SPTree** (Li and Ji, 2014) and **CopyR** (Zeng et al., 2018)). Wherein, **DS-Joint** is an incremental joint framework extracting entities and relations based on structured perceptron and beam-search. **MultiR** is a joint extracting approach for multi-instance learning with overlapping relation types. **CoType** extracts entities and relations by jointly embedding entity mentions, relation mentions, text features, and type labels into two meaningful representations. **SPTree** is a joint learning model that represents both word sequence and dependency tree structures using bidirectional sequential and tree-structured LSTM-RNNs. **CopyR** is a sequence-to-sequence learning framework with a copy mechanism for relation and entity jointly extracting.

(iii) The tagging mechanism based models include **Tagging-BiLSTM** (Zheng et al., 2017) and **Tagging-Graph** (Wang et al., 2018). Wherein, **Tagging-BiLSTM** utilizes a Bi-LSTM-based architecture to capture the context representation of the input sentences through and then uses an LSTM network to decode the tag sequences. **Tagging-Graph** converts the joint extraction task into a directed graph by designing a novel graph scheme.

(iv) RL-based joint extraction models include **HRL**(Takanobu et al., 2018), **JRL**(Zhou et al., 2019), **Seq2SeqRL**(Zeng et al., 2019) and **BIRL**(Wang and Zhang, 2021). Wherein, **HRL** presents a hierarchical RL framework decomposing the whole joint extraction process into a hierarchy of two-level RL policies for relation extraction and entity extraction, respectively. **JRL** consists of two

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 677-688, Harbin, China, August 3 - 5, 2023.

components, including a joint network and a RL agent (which refines the training dataset for anti-noise). **Seq2SeqRL** applies RL strategy into a sequence-to-sequence model to take the extraction order into consideration. **BIRL** proposes a novel bidirectional interaction RL model for jointly extracting entities and relations with both inter-attention and intra-attention.

4.3 Experimental Settings

For a fair comparison, we build our **TERL** implementation for RE and joint extraction task with BERT (Devlin et al., 2019), as BERT-based work has achieved the state-of-the-art performance in RE task. Besides, we adopt BERT as the base encoder for both our **TERL** and other RL-based baselines for a fair comparison. Although GPT is also tested, the experimental trend is consistent. All hyper-parameters are tuned on the validation set. The word vectors are initialized using Word2Vec vectors and are updated during training. DQN encoder (Mnih et al., 2015) with an additional linear layer is introduced here for projecting to the embedding dimension. The main list of hyper-parameters is concluded as follows: Number of layers is 6; Number of attention heads is 8; Embedding dimensionality is 256; Batch size is 512; Context length $K = 30$; Max epochs is 5; Dropout is 0.1; Learning rate is 10^{-4}.

4.4 Performance Summary

F1 results with various labeled data on Relation Extraction (RE) task, are shown in Table 1. Average results over 20 runs are reported, and the best performance is bold-typed. As our work can be viewed as a plug-in unit for RL-based model, the variant model with help of our work is named with suffix "**+TERL**". RL-based methods outperforms all baseline models consistently. We could observe that **+TERL** improve all the RL-based methods. More specifically, compared with the previous SOTA model **GradLRE**, which defeats other models across various labeled data, **+TERL** is also more robust than all the baselines. Considering low-resource RE when labeled data is very scarce, e.g. 5% for SemEval and 3% for TACRED, the improvement from **+TERL** is significant: **+TERL** could achieve an average 3.15% F1 boost compared with **GradLRE**. Moreover, the improvement is still robust when more labeled data can be used (i.e., 30% for SemEval and 15% for TACRED), and the average F1 improvement is 1.15%. Especially, **RDSRE** fall behinds **DualRE** in most cases, while it outperforms **DualRE** when plugged with our **TERL** (i.e., **RDSRE+TERL**). This because the attention mechanism gives our **TERL** an excellent ability of long-term credit assignment, which can capture the effect of actions on rewards in a long sequence. We believe this phenomenon is meaningful and important for document-level RE task. Moreover, a key difference between our TERL and previous RL-based RE SOTA methods, can be concluded that this work dos *not* require policy regularization or conservatism to achieve optimal performance, which is consistent with the observation in (Chen et al., 2021a) and (Zheng et al., 2022). Especially, our speculation is that an algorithm based on time difference learning paradigm learns an approximation function and improves the strategy by optimizing the value function.

Relation and entity joint extraction is a more challenging task, and the proposed Transformer enhanced RL framework can be easily extend to this task. The experimental results on NYT and WebNLG datasets are shown in Table 2. It can be concluded that, the proposed model consistently outperforms all previous SOTA models in most cases, especially RL-base methods. Especially, RL-based methods usually defeats encoder-decoder based methods. E.g., RL-based **HRL** and **JRL** significantly surpass **Tagging-BiLSTM** and **CopyR**. Compared with **HRL**, **JRL** and **BIRL**, the their **+TERL**'s variants improve the F1 score by 3.94%, 3.66% and 4.22% on WebNLG dataset, respectively. This phenomenon shows that, our **TERL**-based variant matches or exceeds the performance of SOTA model-free RL algorithms, even without using dynamic programming. Note that, the behavior of optimizing the learning function in previous work, may unfortunately exacerbate and exploit any inaccuracies in the approximation of the value function, leading to the failure of policy improvement. Due to the fact that the proposed TERL does *not* require explicit optimization with learning functions as the objective, it *avoids* the need for regularization or conservatism, to a certain degree. Moreover, when we represent the distribution of policies, just like sequence understanding, context allows the converter to identify which policies generate actions, thereby achieving better learning and improving training dynamics.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 677-688, Harbin, China, August 3 - 5, 2023.

Table 1: Performance comparisons on Relation Extraction (RE) task (F1).

Model	SemEval			TACRED		
	5%↑	10%↑	30%↑	3%↑	10%↑	15%↑
LSTM(Hochreiter and Schmidhuber, 1997)	0.226	0.329	0.639	0.287	0.468	0.494
PCNN(Zeng et al., 2015)	0.418	0.513	0.637	0.400	0.504	0.525
PRNN(Zhang et al., 2017)	0.553	0.626	0.690	0.391	0.522	0.546
BERT(Devlin et al., 2019)	0.707	0.719	0.786	0.401	0.532	0.556
Self-Training(Rosenberg et al., 2005)	0.713	0.743	0.817	0.421	0.542	0.565
Mean-Teacher(Tarvainen and Valpola, 2017)	0.701	0.734	0.806	0.443	0.531	0.538
DualRE(Lin et al., 2019)	0.744	0.771	0.829	0.431	0.560	0.580
MetaSRE(Hu et al., 2021a)	0.783	0.801	0.848	0.462	0.570	0.589
RDSRE(Qin et al., 2018)	0.729	0.756	0.812	0.422	0.549	0.568
RDSRE+TERL(Ours)	0.787	0.801	0.853	0.435	0.560	0.574
DAGCN(Li et al., 2021)	0.781	0.801	0.838	0.464	0.570	0.587
DAGCN+TERL(Ours)	0.804	0.817	0.846	0.478	0.582	0.593
GradLRE(Hu et al., 2021b)	0.797	0.817	0.855	0.474	0.582	0.599
GradLRE+TERL(Ours)	**0.820**	**0.833**	**0.864**	**0.488**	**0.594**	**0.605**

4.5 Analysis and Discussion

This section investigates whether our TERL variant can remain robust performance on metric of *imitation learning* (like **GradLRE** etc.,) on a *subset* of the dataset. Hence, we adopt baseline **GradLER** which is based on imitation learning, by following the experimental setting of Percentile Behavior Cloning strategy proposed by (Chen et al., 2021a), wherein we run behavior cloning on *only* the top $X\%$ of timesteps in the corresponding dataset, following (Chen et al., 2021a). The Percentile Behavior Cloning variant of **GradLER** is denoted as %**GradLER** here in Table 3. The percentile $X\%$ interpolates between standard behavior cloning ($X = 100\%$) that trains on the complete dataset and only cloning the best observed sequence ($X \approx 0\%$), which in a manner trades off between better generalization by training on more data with training a specialized model that focuses on a subset of the dataset. Table 3 shows experimental results comparing %**GradLRE** to **+TERL**, when the value of X are chosen in $\{10\%, 30\%, 50\%, 100\%\}$. From the experimental results, we conclude that, lower X reduces the performances of **GradLRE**, however **+TERL** successfully exceeds the performance and pulls F1 metric back. Especially, when X is 30, with enhancement from our **TERL**, 30%**GradLRE+TERL** could even defeats 50%**GradLRE**, while 30%**GradLRE** lags behind 50%**GradLRE** obviously. Moreover, 50%**GradLRE+TERL** nearly matches the performance of 100%**GradLRE**. This phenomenon indicates that, the improvement of our **TERL** can be made to the specific subset, after training the distribution of the complete dataset.

Then, to evaluate the importance of accessing previous states, actions, and returns, we discuss the context length K. This is interesting because when using frame stacking, it is usually assumed that the previous state is sufficient for the RL algorithm. Fig. 2 and Fig. 3 is evaluated on RE task (with TACRED dataset and 15% labeled data) and joint extraction task (with WebNLG dataset), respectively. TERL with different K is loaded into baselines **RDSRE**, **DAGCN** and **GradLRE**, as well as baselines **JRL**, **Seq2SeqRL** and **BIRL**. Experimental results show that performance of TERL is significantly worse when K is small (i.e., $K = 1$ or $K = 5$), indicating that past information (i.e., previous states s_t, actions a_t, and returns$\hat{r}_t$) is useful for RE task. Especially, when K becomes small, the performances have fallen off a cliff, even falling behind the original with side effect. Note that, the proposed framework still match the MDP properties when $K = 1$, while the results is worse, which demonstrates the sequence understanding is *highly* context dependent. When $K = 20$ and $K = 30$, **+TERL** defeats the corresponding original comparative baseline and the performances have changed little when K becomes larger. Besides, the context information (i.e., larger K) enables the transformer to figure out which actions are generated, which can lead to higher returns.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 677-688, Harbin, China, August 3 - 5, 2023.

Table 2: Performance comparisons on relation and entity joint extraction task (Precision, Recall, and F1).

Model	NYT			WebNLG		
	Precision↑	Recall↑	F1↑	Precision↑	Recall↑	F1↑
FCM(Kim, 2014)	0.561	0.118	0.193	0.472	0.072	0.124
LINE(Gormley et al., 2015)	0.340	0.251	0.277	0.286	0.153	0.193
MultiR(Hoffmann et al., 2011)	0.344	0.250	0.278	0.289	0.152	0.193
DS-Joint(Yu and Lam, 2010)	0.572	0.201	0.291	0.490	0.119	0.189
CoType(Ren et al., 2017)	0.521	0.196	0.278	0.423	0.175	0.241
SPTree(Li and Ji, 2014)	0.492	0.557	0.496	0.414	0.339	0.357
CopyR (Zeng et al., 2018)	0.569	0.452	0.483	0.479	0.275	0.338
Tagging-BiLSTM(Zheng et al., 2017)	0.624	0.317	0.408	0.525	0.193	0.276
Tagging-Graph(Wang et al., 2018)	0.628	1.632	0.844	0.528	0.194	0.277
HRL(Takanobu et al., 2018)	0.714	0.586	0.616	0.601	0.357	0.432
HRL+TERL(Ours)	0.750	0.604	0.641	0.631	0.368	0.449
JRL(Zhou et al., 2019)	0.691	0.549	0.612	0.581	0.334	0.410
JRL+TERL(Ours)	0.712	0.582	0.613	0.610	0.344	0.425
Seq2SeqRL(Zeng et al., 2019)	0.779	0.672	0.690	0.633	0.599	0.587
Seq2SeqRL+TERL(Ours)	**0.802**	0.692	0.711	0.665	0.617	0.611
BIRL(Wang and Zhang, 2021)	0.756	0.706	0.697	0.660	0.636	0.617
BIRL+TERL(Ours)	0.794	**0.727**	**0.725**	**0.693**	**0.655**	**0.643**

Table 3: Performance comparisons on Percentile Behavior Cloning (F1).

Model	TACRED		
	3%↑	10%↑	15%↑
10%**GradLRE**	0.190	0.233	0.240
10%**GradLRE+TERL**	0.342	0.416	0.424
30%**GradLRE**	0.356	0.437	0.449
30%**GradLRE+TERL**	**0.410**	**0.499**	**0.508**
50%**GradLRE**	0.379	0.466	0.479
50%**GradLRE+TERL**	0.464	0.564	0.575
100%**GradLRE**	0.474	0.582	0.599
100%**GradLRE+TERL**	0.488	0.594	0.605

5 Conclusion

In this work, we try to combine transformers and Reinforcement Learning (RL) based sequence relation extraction (RE), and extend Transformer paradigm to RL. We design a novel framework (TERL) that abstracts RL-based RE as a sequence understanding task, which could leverage the simplicity and scalability of the Transformer-based architecture for understanding textual sequence, as well as the advancements released by pre-training language modeling (such as the BERT/GPT series). Moreover, the proposed framework can also be used as a plug-in unit for any RL-based RE architecture, and be extended to relation and entity joint extraction task. Experimental results show that the proposed TERL framework can improve many state-of-the-art RL-based RE methods.

Acknowledgements

We thank anonymous reviewers for valuable comments. This work is funded by: the National Natural Science Foundation of China (No. U19B2026, 62106243, U22B2601).

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 677-688, Harbin, China, August 3 - 5, 2023.

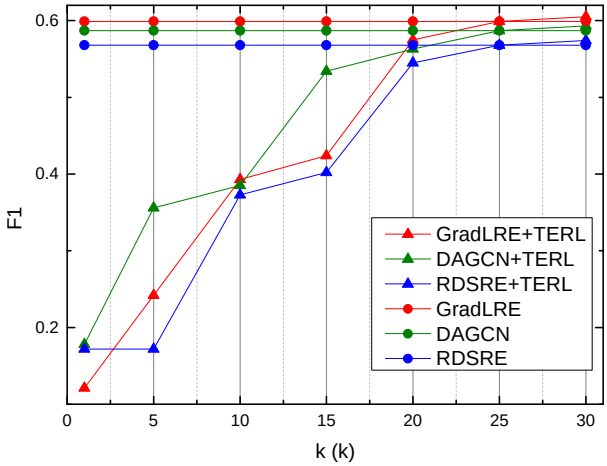

Figure 2: Effect of context length K on RE task.

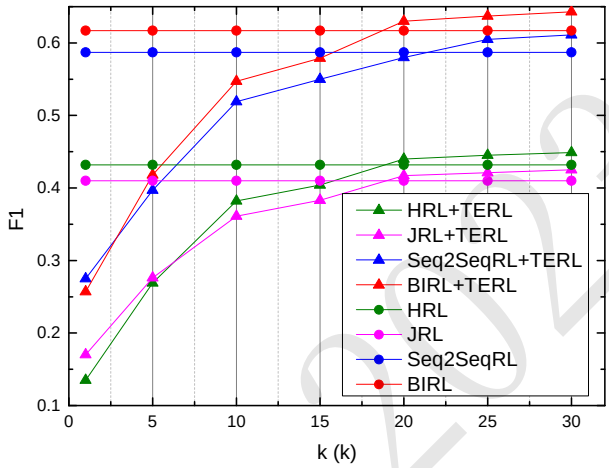

Figure 3: Effect of context length K on joint extraction task.

References

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, P. Abbeel, A. Srinivas, and Igor Mordatch. 2021a. Decision transformer: Reinforcement learning via sequence modeling. In *NeurIPS*.

Yangyi Chen, Jingtong Su, and Wei Wei. 2021b. Multi-granularity textual adversarial attack with behavior cloning. In *EMNLP*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planners. In *ACL*.

Matthew R. Gormley, Mo Yu, and Mark Dredze. 2015. Improved relation extraction with feature-rich compositional embedding models. In *EMNLP*.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*.

Zhijiang Guo, Guoshun Nan, Wei Lu, and Shay B. Cohen. 2020. Learning latent forests for medical relation extraction. In *IJCAI*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In **SEMEVAL*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*.

Xuming Hu, Fukun Ma, Chenyao Liu, Chenwei Zhang, Lijie Wen, and Philip S. Yu. 2021a. Semi-supervised relation extraction via incremental meta self-training. In *EMNLP*.

Xuming Hu, Chenwei Zhang, Yawen Yang, Xiaohe Li, Li Lin, Lijie Wen, and Philip S. Yu. 2021b. Gradient imitation reinforcement learning for low resource relation extraction. In *EMNLP*.

Chia-Chun Hung, Timothy P. Lillicrap, Josh Abramson, Yan Wu, Mehdi Mirza, Federico Carnevale, Arun Ahuja, and Greg Wayne. 2019. Optimizing agent behavior over long time scales by transporting value. *Nature Communications*, 10.

Michael Janner, Qiyang Li, and Sergey Levine. 2021. Reinforcement learning as one big sequence modeling problem. *ArXiv*, abs/2106.02039.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Kimin Lee, Michael Laskin, A. Srinivas, and P. Abbeel. 2021. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In *ICML*.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *ACL*.

Zhixin Li, Yaru Sun, Suqin Tang, Canlong Zhang, and Huifang Ma. 2021. Reinforcement learning with dual attention guided graph convolution for relation extraction. *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 946–953.

Hongtao Lin, Jun Yan, Meng Qu, and Xiang Ren. 2019. Learning dual retrieval module for semi-supervised relation extraction. *The World Wide Web Conference*.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charlie Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature*, 518:529–533.

Karthik Narasimhan, Adam Yala, and Regina Barzilay. 2016. Improving information extraction by acquiring external evidence with reinforcement learning. In *EMNLP*.

Emilio Parisotto and Ruslan Salakhutdinov. 2021. Efficient transformers in reinforcement learning using actor-learner distillation. *ArXiv*, abs/2104.01655.

Emilio Parisotto, H. Francis Song, Jack W. Rae, Razvan Pascanu, aglar Gülehre, Siddhant M. Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, Matthew M. Botvinick, Nicolas Manfred Otto Heess, and Raia Hadsell. 2020. Stabilizing transformers for reinforcement learning. In *ICML*.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018. Robust distant supervision relation extraction via deep reinforcement learning. In *ACL*.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. *ArXiv*, abs/2102.12092.

Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, Tarek F. Abdelzaher, and Jiawei Han. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. *Proceedings of the 26th International Conference on World Wide Web*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML/PKDD*.

Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. 2005. Semi-supervised self-training of object detection models. *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, 1:29–36.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2018. A hierarchical framework for relation extraction with reinforcement learning. In *AAAI*.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 677-688, Harbin, China, August 3 - 5, 2023.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. *ArXiv*, abs/1811.03925.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Yashen Wang and Huanhuan Zhang. 2021. Birl: Bidirectional-interaction reinforcement learning framework for joint relation and entity extraction. In *DASFAA*.

Shaolei Wang, Yue Zhang, Wanxiang Che, and Ting Liu. 2018. Joint extraction of entities and relations based on a novel graph scheme. In *IJCAI*.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. 2020. On layer normalization in the transformer architecture. In *ICML*.

Fuzhao Xue, Aixin Sun, Hao Zhang, and Eng Siong Chng. 2020. Gdpnet: Refining latent multi-view graph for relation extraction. *ArXiv*, abs/2012.06780.

Xiaofeng Yu and Wai Lam. 2010. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In *COLING*.

Vinícius Flores Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David P. Reichert, Timothy P. Lillicrap, Edward Lockhart, Murray Shanahan, Victoria Langston, Razvan Pascanu, Matthew M. Botvinick, Oriol Vinyals, and Peter W. Battaglia. 2019. Deep reinforcement learning with relational inductive biases. In *ICLR*.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *ACL*.

Xiangrong Zeng, Shizhu He, Daojian Zeng, Kang Liu, Shengping Liu, and Jun Zhao. 2019. Learning the extraction order of multiple relational facts in a sentence with reinforcement learning. In *EMNLP/IJCNLP*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *EMNLP*.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. *ArXiv*, abs/1706.05075.

Qinqing Zheng, Amy Zhang, and Aditya Grover. 2022. Online decision transformer. In *ICML*.

Xin Zhou, Luping Liu, Xiaodong Luo, Haiqiang Chen, Linbo Qing, and Xiaohai He. 2019. Joint entity and relation extraction based on reinforcement learning. *IEEE Access*, 7:125688–125699.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 677-688, Harbin, China, August 3 - 5, 2023.