UniRel: Unified Representation and Interaction for Joint Relational Triple Extraction

Wei Tang [1,2], **Benfeng Xu** [1,2], **Yuyue Zhao** [1,2], **Zhendong Mao** [1,2], **Yifeng Liu** [3]
Yong Liao [1,2] *, **Haiyong Xie** [1,2]
[1] University of Science and Technology of China, Anhui, China
[2] CCCD Key Lab of Ministry of Culture and Tourism, Anhui, China
[3] National Engineering Laboratory for Risk Perception and Prevention (RPP), Beijing, China
{weitang, benfeng, yyzha0}@mail.ustc.edu.cn, zdmao@ustc.edu.cn, liuyifeng3@cetc.com.cn
{yliao, hxie}@ustc.edu.cn

Abstract

Relational triple extraction is challenging for its difficulty in capturing rich correlations between entities and relations. Existing works suffer from 1) heterogeneous representations of entities and relations, and 2) heterogeneous modeling of entity-entity interactions and entity-relation interactions. Therefore, the rich correlations are not fully exploited by existing works. In this paper, we propose UniRel to address these challenges. Specifically, we unify the representations of entities and relations by jointly encoding them within a concatenated natural language sequence, and unify the modeling of interactions with a proposed Interaction Map, which is built upon the off-the-shelf self-attention mechanism within any Transformer block. With comprehensive experiments on two popular relational triple extraction datasets, we demonstrate that UniRel is more effective and computationally efficient. The source code is available at https://github.com/wtangdev/UniRel.

1 Introduction

Relational Triple Extraction (RTE) aims to identify entities and their semantic relations jointly. It extracts structured triples in the form of <*subject-relation-object*> from raw texts in an end-to-end manner, and is a crucial task towards automatically constructing large-scale knowledge bases (Nayak et al., 2021).

Early works try to solve RTE tasks in a pipelined fashion involving two sub-tasks (Zelenko et al., 2003; Chan and Roth, 2011), where entities are firstly recognized, and relations are then assigned for each extracted entity pair. Therefore, such methods fail to capture the implicit correlation between these two isolated sub-tasks and are thus prone to propagated errors (Li and Ji, 2014).

Instead, many researchers (Miwa and Bansal, 2016; Zheng et al., 2017; Zeng et al., 2018) seek to jointly extract <s-r-o> triples in an end-to-end manner. For example, Wei et al. (2020) propose a cascaded network that identifies subjects first and then recognizes corresponding objects for relations. Zheng et al. (2021a) decompose RTE into three subtasks. Wang et al. (2020) and Shang et al. (2022) extract relational triples in one stage to eliminate exposure bias. In general, these works accomplish end-to-end extraction by various ways to factorize and re-assemble the label space of relational triples. However, very often, there exist rich informative correlations between entities and relations, and these correlations can hardly be captured by the superficial constraints applied in label space only. Specifically, existing works fall short of two aspects: 1) heterogeneous representations between entities and relations; 2) ignorance of interactive dependencies between entity-level interactions and relation-level interactions.

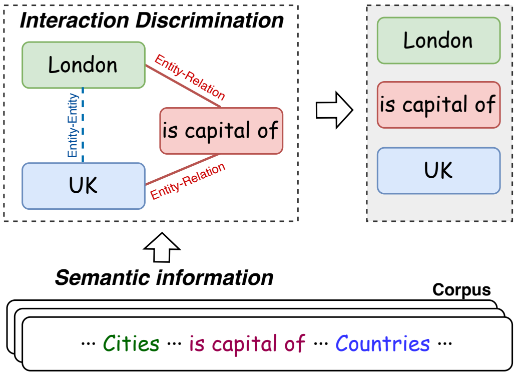


Figure 1: We leverage semantic information to unify the representation of entities and relations. Relational triples are extracted by modeling the entity-entity interaction (blue dashed line) and entity-relation interaction (red solid line) in a unified way.

For representations, existing works mainly focus on how to better capture the contextual information of entities, while ignoring the equally important semantic meaning of relations. Generally, rela-

[*] Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 7087–7099
December 7-11, 2022 ©2022 Association for Computational Linguistics

tions are simply represented as atomic label ids indicating a specific dimension in the newly initialized classifier (Wei et al., 2020; Wang et al., 2020; Zheng et al., 2021a), resulting in its heterogeneity with language model augmented entity representations. Such heterogeneity prevents models from capturing the intrinsic correlations between entities and relations in the semantic space. For instance, from the semantic meaning of the relation `is_capital_of`, we can infer that triples involving are related to locations: the subject is supposed to be a city, while the object should be a country. We argue that it is important to build unified representations for both entities and relations.

For interactions, we exemplify the interdependence between the entity-entity interactions and entity-relation interactions in Figure 1. We can easily determine that *London* and *UK* are correlated in the prerequisite of given the interactions of (*London*-`is_capital_of`) and (*UK*-`is_capital_of`). However, existing works either enumerate all entity-relation-entity triples (Wang et al., 2020; Shang et al., 2022), which suffer from huge prediction space, or model them with separate modules (Wei et al., 2020; Zheng et al., 2021a; Li et al., 2021) respectively for named entity recognition (NER) and relation extraction (RE), resulting in neglect of interdependencies. In general, the absence of unified modeling of interactions limits these methods from fully utilizing their interdependencies for better extraction.

In this paper, we propose **UniRel** with ***Unified Representation*** and ***Interaction*** to resolve both the heterogeneity of representations and the absence of interaction dependencies. We first encode both relation and entity into meaningful sequence embeddings to construct unified representations. Based on the semantic definition, candidate relations are first converted to natural language texts, and form a consecutive sequence together with the input sentence. We then apply a Transformer-based Pre-trained Language Model (PLM) to encode the sequence while intrinsically capturing their informative correlations. We then propose a novel solution for ***Unified Interaction***, where we simultaneously model the entity-entity interactions and entity-relation interactions in one single `Interaction Map` by leveraging the off-the-shelf self-attention mechanism inside Transformer. Besides, benefiting from the design of Interaction Map, UniRel preserves the advantages for end-to-end extraction and is superior in computational efficiency.

We conduct comprehensive experiments on two popular datasets: NYT (Riedel et al., 2010) and WebNLG (Gardent et al., 2017), and achieve a new state-of-the-art. We summarize our contributions as follows:

- We propose unified representations of entities and relations by jointly encoding them within a concatenated natural language sequence, which fully exploits their contextualized correlations and leverages the semantic knowledge learned from LM pre-training.

- We propose unified interactions to capture the interdependencies between entity-entity interactions and entity-relation interactions. This is innovatively achieved by the proposed Interaction Map built upon the off-the-shelf self-attention mechanism within any Transformer block.

- We show that UniRel achieves the new state-of-the-art for RTE tasks, while preserving the superiority of being end-to-end and computationally efficient.

2 Related Work

Early works (Zelenko et al., 2003; Chan and Roth, 2011) apply pipeline approaches that divide relational triple extraction into two isolated sub-tasks: first do named entity recognition to extract all entities, and then apply relation extraction to identify relations for each entity pair. However, these methods always suffer from error propagation for failing to capture the implicit correlation between these two isolated sub-tasks.

To tackle these issues, recent researches focus on jointly extracting entities and relations. Previous feature-based joint models (Yu and Lam, 2010; Miwa and Sasaki, 2014; Li and Ji, 2014; Ren et al., 2017) require complex feature engineering and heavily depend on NLP tools. Researchers propose neural network-based joint models to eliminate hand-craft features. Miwa and Bansal (2016) propose a model to jointly learn entities and relations through parameter sharing. Zheng et al. (2017) transform RTE into a sequence tagging problem, which unifies the annotation role of entities and relations.

Despite their success, most models cannot deal with complex scenarios where one sentence consists of multiple overlapping relational triples sharing a single entity (SingleEntityOverlap, SEO) or an entity pair (EntityPairOverlap, EPO). To handle the problem, researchers (Zeng et al., 2018, 2019; Nayak and Ng, 2020; Ye et al., 2021) propose generative models that view triple as a token sequence. Some works (Wang et al., 2020; Ren et al., 2021; Shang et al., 2022) introduce methods to extract triples in one-stage but suffer from huge prediction space. Other researchers (Wei et al., 2020; Yuan et al., 2020; Zheng et al., 2021b; Li et al., 2021; Wu and Shi, 2021) decompose RTE into different sub-tasks, but learn the interaction between sub-tasks only by input sharing, or falling into the cascade error. PFN (Yan et al., 2021) proposes a partition filter network to fuse the task representation of NER and RE, but still models entity-entity interactions and entity-relation interactions with separate modules. In this work, UniRel unifies the modeling of the two kinds of interactions in one single Interaction Map to fully capture their interdependencies and is superior in computational efficiency.

More recently, Xu et al. (2022) propose EmRel that explicitly introduce relation representation to leverage the rich interactions across relations, entities, and context. However, it still suffers from heterogeneity between entities and the newly initialized embeddings of relations. Some approaches (Han et al., 2021; Chen and Li, 2021; Chen et al., 2022) introduce prompt-tuning to extract relation with semantic information. They transform the relation extraction task into a masked language modeling problem. However, such methods focus on the simple scenario of sentence-level relation classification without capturing the correlations between entities and relations. In terms of technical designs, SSAN (Xu et al., 2021) also delves into the self-attention layer within the Transformer to model structural interaction, but it makes extra adaptions for standard self-attention mechanism and focuses on document-level RE tasks. Compared to these approaches, our work aims to extract relational triples in complex scenarios where rich intrinsic correlations exist between entities and relations. In this paper, we unify the representations and interactions to fully exploit the correlations to extract entities and relations jointly.

3 Methodology

In this section, we present our model in detail. We first introduce the problem formulation in Section 3.1. Then, we introduce the Unified Representation and the Unified Interaction in Section 3.2 and Section 3.3, respectively. Finally, we present the details of training and decoding in Section 3.4.

3.1 Problem Formulation

Given a sentence $X = \{x_1, x_2, \cdots, x_N\}$ with N tokens, the goal of joint relational triple extraction is to identify all possible triples $T = [(s_l, r_l, o_l)]_{l=1}^{L}$ from X, where s_l, r_l, p_l represent the subject, the object, and their relation, respectively, and L is the number of triples. The subject and object are entity mentions $E = \{e_1, e_2, \cdots, e_k\}$ from sentence X, k is the number of entities. The relation is from pre-defined relation set $R = \{R_1, R_2, \cdots, R_M\}$ with M types. Note that entities and relations might be shared among triples.

3.2 Unified Representation

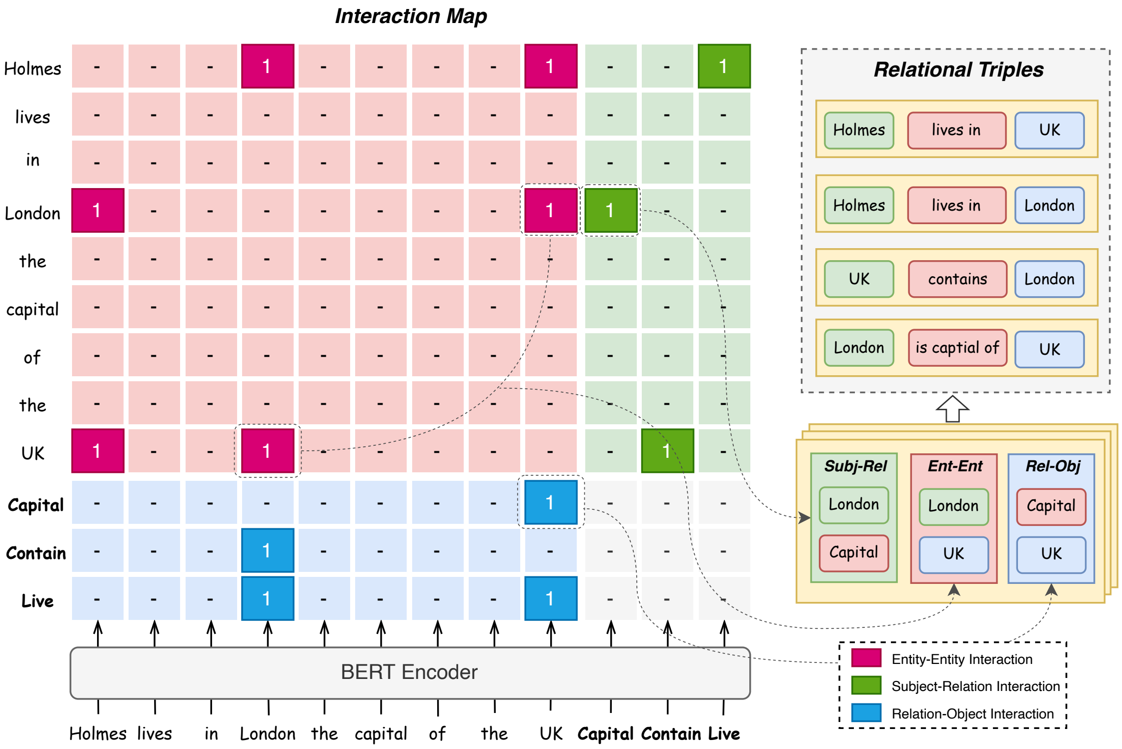
We first convert relations in the schema to natural language texts to represent them in the same form as the input sentence. For clarity, the conversion is performed through a verbalizer with human-picked words. The relation word is basically the most informative word within the label name that preserves its semantics, for example, "founders" for relation "`/business/company/founders`". We then input the concatenation of the input sentence and the natural language texts of relations to a Transformer-based PLM encoder. We use BERT (Devlin et al., 2019) as the PLM in this work. The inputs are then transferred to a sequence of input embeddings by searching the corresponding ids from the embedding table.

$$T = \text{Concat}(T_s, T_p) \tag{1}$$

$$H = E[T] \tag{2}$$

where $H \in \mathbb{R}^{(N+M)\times d_h}$ is the input embedding vector. d_h is the embedding size. T_s and T_p are the input ids of the input sentence and the relations, respectively. E is the embedding table in BERT.

After obtaining the input embeddings, the encoder captures the correlations between each input word with the self-attention mechanism. To be specific, Transformer-based PLMs comprise stacked Transformer (Vaswani et al., 2017) layers consisting of multiple attention heads. Each head applies three separate linear transformations to transform the input embeddings H to query, key, and value vectors Q, K, V, and then computes the attention



Interaction Map

	Holmes	lives	in	London	the	capital	of	the	UK	Capital	Contain	Live
Holmes	-	-	-	1	-	-	-	-	1	-	-	1
lives	-	-	-	-	-	-	-	-	-	-	-	-
in	-	-	-	-	-	-	-	-	-	-	-	-
London	1	-	-	-	-	-	-	-	1	1	-	-
the	-	-	-	-	-	-	-	-	-	-	-	-
capital	-	-	-	-	-	-	-	-	-	-	-	-
of	-	-	-	-	-	-	-	-	-	-	-	-
the	-	-	-	-	-	-	-	-	-	-	-	-
UK	1	-	-	1	-	-	-	-	-	-	1	-
Capital	-	-	-	-	-	-	-	-	1	-	-	-
Contain	-	-	-	1	-	-	-	-	-	-	-	-
Live	-	-	-	1	-	-	-	-	1	-	-	-

BERT Encoder
Holmes lives in London the capital of the UK **Capital Contain Live**
Relational Triples
Holmes | lives in | UK
Holmes | lives in | London
UK | contains | London
London | is captial of | UK
Subj-Rel: London, Capital
Ent-Ent: London, UK
Rel-Obj: Capital, UK
Entity-Entity Interaction
Subject-Relation Interaction
Relation-Object Interaction

Figure 2: We input the concatenation of the input sentence and the natural language texts of relations (in bold). The Interaction Map is learned from the attention map inside the 12th layer of BERT Encoder, which consists of Entity-Entity Interaction (red rectangle) and Entity-Relation Interaction (green rectangle for subject and blue rectangle for object). Relational triples are extracted intuitively from the map.

weights between all pairs of words by Softmax-normalized dot production of Q and K, which then is fused with V as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_h}})V. \quad (3)$$

Each Transformer layer generates token embeddings from the previous layer's output with the self-attention mechanism. We denote the H_i as the output of the i-the Transformer layer. As we take both entities and relations into the input embedding H, the H_i, encoded by such a deep Transformer network, fully captures the rich intrinsic correlations between entities and relations. The above steps make the representations of entities and relations unified into one semantic embedding vector H_i with rich correlational information.

3.3 Unified Interaction

The interactions between the triple elements, *Entity-Entity Interaction* and *Entity-Relation Interaction*, can be directly used to extract relational triples. As shown in Figure 2, Triple (*London*, `is_capital_of`, *UK*) can be determined if knowing the interactions between (*London-UK*), (*London*-`is_capital_of`), and (*UK*-`is_capital_of`). Motivated by this, we design an Interaction Map to model the two kinds of interactions simultaneously.

3.3.1 Entity-Entity Interaction

Entity-entity interaction is defined for identifying entity pairs that can be used to form valid relational triples. Given two entities e_a and e_b from sentence X, we regard entity pair (e_a, e_b) are interacted only when there exists relation r that can be formed as valid triples together with them, and both (e_a, r, e_b) and (e_b, r, e_a) are allowed. For example, in Figure 2, (*Holmes* - *London*), as well as (*London* - *Holmes*), are supposed to be interacted for the existing triple (*Holmes*, `lives in`, *London*), while unrelated for (*Holmes* - *capital*) since no valid triple consists of them. Formally, we define the entity-entity interaction indicator function $I_e(\cdot)$ as follows:

$$I_e(e_a, e_b) = \begin{cases} \textit{True} & (e_a, r, e_b) \in T \text{ or} \\ & (e_b, r, e_a) \in T, \exists r \in R\ , \\ \textit{False} & \text{otherwise} \end{cases} \quad (4)$$

$I_e(e_b, e_a) = I_e(e_a, e_b)$, as entity-entity interaction is symmetrical.

3.3.2 Entity-Relation Interaction

Entity-relation interaction recognizes correlated entities for each relation. Given a relation r, we regard entity e as interacting with r when existing triples consisting of e as either subject or object and r as the relation. As the relation is directional, we define entity-relation interaction asymmetrically to distinguish subject entities and object entities, as shown in the upper right part (Subject-Relation) and lower left part (Relation-Object) of the map in Figure 2, respectively. For instance, the interaction value of (*London* - `is_capital_of`) is supposed to be *True* because of the valid triple (*London*, `is_capital_of`, *UK*), while *False* for (*UK* - `is_capital_of`) since it is impossible for *UK* to be the subject of the relation `is_capital_of`. We formally define the indicator function $I_r(\cdot)$ of entity-relation interaction as follows:

$$I_r(e, r) = \begin{cases} \mathit{True} & (e, r, o) \in T, \exists o \in E \\ \mathit{False} & \text{otherwise} \end{cases}, \quad (5)$$

$$I_r(r, e) = \begin{cases} \mathit{True} & (s, r, e) \in T, \exists s \in E \\ \mathit{False} & \text{otherwise} \end{cases}, \quad (6)$$

where $I_r(e, r)$ and $I_r(r, e)$ are defined for identifying entity e as subject and object for relation r, respectively.

3.3.3 Interaction Discrimination

Transformer layers bring powerful deep correlation capturing ability to BERT. Therefore, As shown in Figure 2, we comprise the two kinds of interactions into one single Interaction Map, which is in the same form as the attention map computed by the Transformer layer. Then we directly take the last Transformer layer of BERT for Interaction Discrimination. As the Interaction Map is not restricted to a normalized matrix, after obtaining Q, K from H_{11}, the embeddings generated by the last layer of BERT, we average the dot production of Q and K of all heads and directly apply the sigmoid function to obtain the results. The detailed operations are as follows:

$$\mathbf{I} = \text{sigmoid}\Big(\frac{1}{T}\sum_{t}^{T}\frac{Q_t K_t^T}{\sqrt{d_h}}\Big), \quad (7)$$

where $\mathbf{I} \in \mathbb{R}^{(N+M)(N+M)}$ is the interaction matrix corresponding to the Interaction Map. T is the number of heads. W_t^Q and W_t^K are trainable weights. We consider $\mathbf{I}(\cdot)$ valid when the value of $\mathbf{I}(\cdot)$ exceeds threshold σ.

UniRel captures the interactive dependencies as the entity-entity interactions and entity-relation interactions are simultaneously modeled in one single Interaction Map. Besides, with Unified Interaction, the prediction space is narrowed down to $O((N+M)^2)$, which is much smaller than the most recent work, OneRel, which predicts triples in the complexity of $O(N \times M \times N)$.

3.4 Training and Decoding

The binary cross entropy loss is used for training:

$$\mathcal{L} = -\frac{1}{(N+M)^2}\sum_{i}^{N+M}\sum_{j}^{N+M}(\mathbf{I^*}_{i,j}\log\mathbf{I}_{i,j} + (1-\mathbf{I^*}_{i,j})\log(1-\mathbf{I}_{i,j})), \quad (8)$$

where $\mathbf{I}^*$ is the ground truth matrix of the Interaction Map.

For decoding, we first recognize all valid subject entities and object entities for each relation from entity-relation interactions I_r (the green box and blue box in the lower right part of Figure 2). Then we enumerate all candidate entity pairs for each relation pruned by entity-entity interactions I_e (the red box in the lower right part of Figure 2).

Such a decoding method can address the complex scenarios with overlapping patterns containing *EntityPairOverlap* (EPO) and *SingleEntityOverlap* (SEO) for all arrangements of the entities and relations are taken into account. As shown in Figure 2, the extracted triples contain SEO triples: (*Holmes*, `lives_in`, *UK*) and (*Holmes*, `lives_in`, *London*), and the EPO triples: (*UK*, `contains`, *London*) and (*London*, `is_capital_of`, *UK*).

4 Experiments

4.1 Datasets and Evaluation

We evaluate the proposed method on two widely used benchmark datasets NYT (Riedel et al., 2010) and WebNLG (Gardent et al., 2017). NYT dataset is produced by distant supervision from New York Times articles which has 24 predefined relations. WebNLG was first created for natural language generation task and is adapted to relational triple extraction by Zeng et al. (2018), which has 171 predefined relations. The statistics of the datasets are shown in Table 2. We evaluate our method on standard data splitting and report the standard

Model	NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1
NovelTagging (Zheng et al., 2017)	62.4	31.7	42.0	52.5	19.3	28.3
CopyRE (Zeng et al., 2018)	61.0	56.6	58.7	37.7	36.4	37.1
GraphRel (Fu et al., 2019)	63.9	60.0	61.9	44.7	41.1	42.9
OrderCopyRE (Zeng et al., 2019)	77.9	67.2	72.1	63.3	59.9	61.6
CasRel$_{BERT}$ (Wei et al., 2020)	89.7	89.5	89.6	93.4	90.1	91.8
TPlinker$_{BERT}$ (Wang et al., 2020)	91.3	92.5	91.9	91.7	92.0	91.9
PRGC$_{BERT}$ (Zheng et al., 2021a)	93.3	91.9	92.6	94.0	92.1	93.0
R-BPtrNet$_{BERT}$ (Chen et al., 2021)	92.7	92.5	92.6	93.7	92.8	93.3
PFN (Yan et al., 2021)	-	-	92.4	-	-	93.6
TDEER$_{BERT}$ (Li et al., 2021)	93.0	92.1	92.5	93.8	92.4	93.1
GRTE$_{BERT}$ (Ren et al., 2021)	92.9	93.1	93.0	93.7	94.2	93.9
EmRel (Xu et al., 2022)	91.7	92.5	92.1	92.7	93.0	92.9
OneRel$_{BERT}$ (Shang et al., 2022)	92.8	92.9	92.8	94.1	94.4	94.3
UniRel	**93.5**	**94.0**	**93.7**	**94.8**	**94.6**	**94.7**
UniRel$_{unused}$	93.1	93.2	93.1	75.1	68.6	71.7
UniRel$_{separate}$	92.6	93.7	93.1	93.5	94.4	93.9

Table 1: Main results. The highest scores are in bold.

micro Precision (Prec.), Recall (Rec.), and F1-score on test set following the same setting as Zeng et al. (2019).

Our model is implemented based on Pytorch. We optimize the parameters by Adam (Kingma and Ba, 2015) using learning rates 3e-5 and 5e-5 for NYT and WebNLG, respectively. The learning rates are searched in {3e-5, 5e-5, 7e-5}. We also conduct weight decay (Loshchilov and Hutter, 2019) with a rate of 0.01. The batch size is 24/6 for NYT/WebNLG and trained for 100 epochs. We use cased BERT-base [1] with 108M parameters as the PLM and set the max length of the input sentence to 100 to keep in line with previous work. The size of attention head d_h is 64. The threshold σ is set as 0.5. We tune the parameters on the development set. Our experiments are conducted on one NVIDIA V100 32GB GPU.

Dataset	Train	Test	Overlapping Pattern			
			Normal	SEO	EPO	SOO
NYT	56195	5000	3266	1297	978	45
WebNLG	5019	703	245	457	26	84

Table 2: Statistics of evaluation datasets. Overlapping patterns are counted on test set.

4.2 Results

For comparison, We employed twelve strong models as baselines consisting of the SOTA models PRGC (Zheng et al., 2021a), PFN (Yan et al., 2021), TDEER (Li et al., 2021), GRTE (Ren et al., 2021) and OneRel (Shang et al., 2022). We take the experiment results from the original papers of these baselines directly.

Table 1 shows the results of our model against other baseline methods on all datasets. Many previous baselines achieve F1-score of over 90% on both datasets, especially on WebNLG, which already exceed human-level performance. UniRel achieves +0.7% and +0.4% improvements over F1-scores on NYT and WebNLG and outperforms all the baselines in terms of all the evaluation metrics, which shows the superiority of our model.

To further study the ability to handle the overlapping problem and extracting multiple triples, following previous works (Wei et al., 2020; Wang et al., 2020; Zheng et al., 2021a; Shang et al., 2022), we conduct further experiments on different types subsets of NYT.

As shown in Table 3, the results indicate the effectiveness of our model in complex scenarios. Our model exceeds almost all the baselines in the Normal class and three overlapping patterns. Especially in SEO and EPO, the most common over-

[1] https://huggingface.co/bert-base-cased

Model	Normal	SEO	EPO	SOO	L= 1	L= 2	L= 3	L= 4	L≥ 5
CasRel	87.3	91.4	92.0	77.0	88.2	90.3	91.9	94.2	83.7
TPlinker	90.1	93.4	94.0	90.1	90.0	92.8	93.1	96.1	90.0
PRGC	91.0	94.0	94.5	81.8	91.1	93.0	93.5	95.5	93.0
R-BPtrNet	90.4	94.4	**95.2**	-	89.5	93.1	93.5	**96.7**	91.3
PFN	90.2	**95.3**	94.1	-	90.5	92.9	93.7	96.3	92.6
TDEER	90.8	94.1	94.5	-	90.8	92.8	94.1	95.9	92.8
GRTE	91.1	94.4	95.0	-	90.8	93.7	94.4	96.2	93.4
OneRel	90.6	95.1	94.8	**90.8**	90.5	93.4	93.9	96.5	**94.2**
UniRel	$\mathbf{91.6}_{\pm0.3}$	$\mathbf{95.3}_{\pm0.2}$	$\mathbf{95.2}_{\pm0.1}$	$89.8_{\pm3.6}$	$\mathbf{91.5}_{\pm0.3}$	$\mathbf{94.3}_{\pm0.2}$	$\mathbf{94.5}_{\pm0.3}$	$96.6_{\pm0.2}$	$\mathbf{94.2}_{\pm0.8}$

Table 3: F1-score on sentences with different overlapping patterns and different triple numbers. L is the number of triples in one sentence. All the compared models are implemented with BERT. We report the average results of UniRel of the five runs with different random seeds. The highest scores are in bold.

lapping situations, our model achieves the highest performance, and the results are robust, which demonstrates the advantage of UniRel in processing overlapping triples. It can be seen that our model also makes improvements for almost all kinds of sentences regarding the number of triples. From simple situation (L $=$ 1) to complex case (L $=$ 3), UniRel still brings improvements, which shows the robustness of our model. In general, this experiment shows the power of our model in complex scenarios. We attribute the effectiveness to the captured rich interactions between entities and relations by the introduced Interaction Map, which is essential for solving the complex overlapping triple problem.

5 Analysis

5.1 Ablation Study

In this section, we conduct ablation experiments on NYT and WebNLG datasets to study the effectiveness of the proposed Unified Representation and Unified Interaction as reported in Table 1.

5.1.1 Effect of the Unified Representation

To study the effectiveness of Unified Representation, instead of assigning meaningful words for each relation, we use the placeholder `[unused]` of BERT to represent relations as marked as UniRel$_{unused}$ in Table 1. Same as meaningless label ids, the embeddings of `[unused]` tokens are randomly initialized at the fine-tuning stage without augmenting the meaningful semantic information learned from pre-training. We can see a performance decay in terms of all the evaluation metrics on both datasets without Unified Representation, which indicates the importance of the semantic information for relational triple extraction.

We also notice the significant performance decrease of UniRel$_{unused}$ on the WebNLG dataset. We think it is because the WebNLG dataset has much fewer training data but defines far more relations compared to the NYT dataset. Such a contradiction makes many relations have few samples for training in the WebNLG dataset. And it is hard for a model to learn the deep semantic information with few examples from zero.

To validate our assumption, we further analyze the performance of UniRel and UniRel$_{unused}$ on relations with different orders of magnitude samples in training data on the WebNLG dataset. As shown in Figure 3, UniRel$_{unused}$ performs well on relations with much samples ($\geq$ 1000), but is limited when the number of samples decreases, which confirms our assumption. In contrast, UniRel maintains a good performance level in the face of different numbers of samples. Especially with extremely few samples ($\leq$ 10), UniRel performs at the same level as much samples ($\geq$ 1000) , which further demonstrates the effectiveness of Unified Representation.

5.1.2 Effect of the Unified Interaction

To study the influence of Unified Interaction, we take the relation sequence out of the input sentence to model the two kinds of interactions in a separate manner, and denoted as UniRel$_{separate}$ in Table 1. Specifically, we first obtain the sequence embeddings of the input sentence and the natural language texts of relations severally with the same BERT encoder. Then we apply two transform layers to get the query and key of the concatenation of

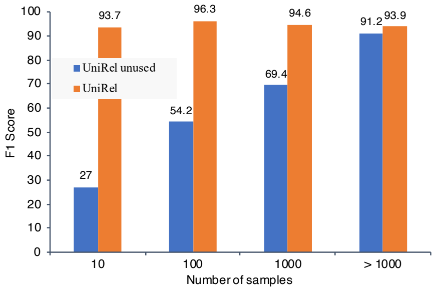


Figure 3: F1-score on relations with different orders of magnitude samples in training set for UniRel (in Orange) and UniRel$_{unused}$ (in Blue). 10/100/1000 means relations with less than or equal to 10/100/1000 samples.

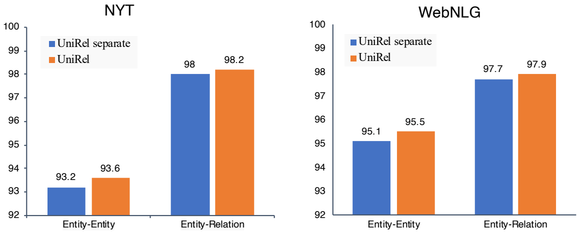


Figure 4: F1-score on Entity-Entity Interaction and Entity-Relation Interaction for UniRel (in Orange) and UniRel$_{separate}$ (in Blue).

the two embeddings. Finally, the Interaction Map is outputted by doing dot production of query and key.

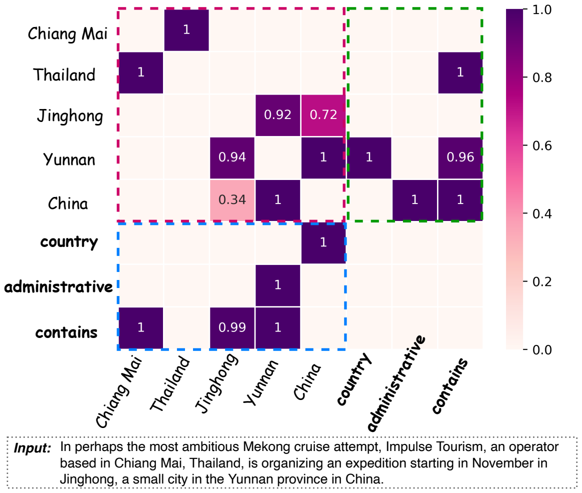
As UniRel$_{separate}$ takes entity and relation as individual inputs, the deep Transformer network can only independently model the correlations inside entity pair and relation pair without capturing the interdependencies between entity-entity interactions and entity-relation interactions. As shown in Table 1, UniRel$_{separate}$ has marked performance degradation on both datasets compared to UniRel, which demonstrates the unified interaction's effectiveness. We further analyze the performances of UniRel and UniRel$_{separate}$ on different interaction types. As shown in Figure 4, not only entity-relation, the F1-score of entity-entity also decreases without simultaneously modeling the interactions, which proves the interdependencies between the two kinds of interactions, and UniRel takes benefits from modeling them in a unified way.

5.2 Computational Efficiency

Table 4 shows the comparison results of the computational efficiency between UniRel and some recent high-performance models. We report training and inference time on both NYT and WebNLG datasets. In this experiment, we follow previous works and set the batch size to 6/1 for training/inference. All the compared models are tested in the same hardware environment as declared in Section 4.1.

Model	Training Time		Inference Time	
	NYT	WebNLG	NYT	WebNLG
CasRel	1142	105	35	37
TPLinker	2951	810	48	57
PRGC	3632	498	13	13
OneRel	2998	186	20	21
UniRel	967	119	12	14

Table 4: Computational Efficiency. Training time represents the time (second) needed to train one epoch. Inference time represents the average time (millisecond) to predict one sample.



Input: In perhaps the most ambitious Mekong cruise attempt, Impulse Tourism, an operator based in Chiang Mai, Thailand, is organizing an expedition starting in November in Jinghong, a small city in the Yunnan province in China.

Figure 5: Visualization of Interaction Map with input sentence sampled from NYT. Relations are in bold.

UniRel shows a conspicuous computational efficiency performance in both training and inference time. Specifically, Compared to the SOTA model OneRel, on the NYT dataset, UniRel obtains 3× and 1.7× faster in the stage of training and inference, respectively. We think the reason is that UniRel $(O(N+M)^2)$ has a smaller prediction space than OneRel $(O(N \times M \times N))$. Although CasRel performs similarly to ours regarding training time, UniRel obtains more than 2.6× speedups in the inference time. We attribute the efficiency to the design of the Interaction Map, which allows UniRel to narrow down the prediction space and directly leverage the off-the-shelf self-attention mechanism within the Transformer block.

5.3 Visualization

We visualize the Interaction Map to see how it works for relational triple extraction. As shown in Figure 5, the red box represents the entity-entity interaction. The blue box and green box represent the entity-relation interaction for the subject and the object, respectively. From the map, we can extract all six relational triples: (*Yunnan*, `country`, *China*), (*China*, `administrative_divisions`, *Yunnan*), (*Thailand*, `contains`, *Chiang Mai*), (*Yunan*, `Contains`, *Jinghong*), (*China*, `Contains`, *Jinghong*), and (*China*, `Contains`, *Yunnan*).

6 Conclusion

In this work, we propose UniRel to fully leverage the rich correlations between entities and relations by resolving the heterogeneity. Unified Representation eliminates the representation's heterogeneity by encoding both entity and relation into meaningful sequence embeddings. Unified Interaction eliminates the interaction's heterogeneity by simultaneously modeling entity-entity interactions and entity-relation interactions in one single Interaction Map. UniRel produces significant improvements over competitive baselines. We give a comprehensive analysis to further justify our design.

Limitations

There are two limitations we want to discuss in this section:

- First, for clarity, we select the corresponding words for each relation in a manual way, which would be sophisticated for schema with much relations. We will next to try to design an auto-verbalizer for relations.

- Second, the evaluation benchmarks (NYT and WebNLG) for joint relational triple extraction are produced with much annotated training data, which is expensive for real application. The performance of our model in low-resource scenario needs to be validated. However, existing benchmarks for low-resource (Han et al., 2018; Gao et al., 2019) are limited to the simple scenario of sentence-level relation classification. We would like to explore the the idea of unified representation and unified interaction towards joint RTE in low resource scenario in the future.

Acknowledgements

This work is supported by the National Key Research and Development Program of China (2021YFC3300500).

References

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 551–560, Portland, Oregon, USA. Association for Computational Linguistics.

Chih-Yao Chen and Cheng-Te Li. 2021. ZS-BERT: Towards zero-shot relation extraction with attribute representation learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3470–3479, Online. Association for Computational Linguistics.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 2778–2788, New York, NY, USA. Association for Computing Machinery.

Yubo Chen, Yunqi Zhang, Changran Hu, and Yongfeng Huang. 2021. Jointly Extracting Explicit and Implicit Relational Triples with Reasoning Pattern Enhanced Binary Pointer Network. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5694–5703, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT 2019*, pages 4171–4186.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. GraphRel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418, Florence, Italy. Association for Computational Linguistics.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. FewRel 2.0: Towards more challenging few-shot relation classification. 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 6250–6255, Hong Kong, China. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *ACL 2017*, pages 179–188.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. *arXiv preprint arXiv:2105.11259*.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR 2015*.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.

Xianming Li, Xiaotian Luo, Chenghao Dong, Daichuan Yang, Beidi Luan, and Zhen He. 2021. TDEER: An efficient translating decoding schema for joint extraction of entities and relations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8055–8064, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR 2019*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *EMNLP 2014*, pages 1858–1869.

Tapas Nayak, Navonil Majumder, Pawan Goyal, and Soujanya Poria. 2021. Deep neural approaches to relation triplets extraction: A comprehensive survey. *Cognitive Computation*, 13(5).

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *AAAI 2020*, pages 8528–8535.

Feiliang Ren, Longhui Zhang, Shujuan Yin, Xiaofeng Zhao, Shilei Liu, Bochao Li, and Yaduo Liu. 2021. A novel global feature-oriented relational triple extraction model based on table filling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2646–2656, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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. In *WWW 2017*, pages 1015–1024.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML PKDD 2010*, volume 6323 of *Lecture Notes in Computer Science*, pages 148–163.

Yu-Ming Shang, Heyan Huang, and Xianling Mao. 2022. Onerel: Joint entity and relation extraction with one module in one step. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11285–11293.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS 2017*, pages 5998–6008.

Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020. TPLinker: Single-stage Joint Extraction of Entities and Relations Through Token Pair Linking. In *COLING 2020*, pages 1572–1582.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020. A novel cascade binary tagging framework for relational triple extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1476–1488, Online. Association for Computational Linguistics.

Hui Wu and Xiaodong Shi. 2021. Synchronous dual network with cross-type attention for joint entity and relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2769–2779, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Benfeng Xu, Quan Wang, Yajuan Lyu, Yabing Shi, Yong Zhu, Jie Gao, and Zhendong Mao. 2022. EmRel: Joint representation of entities and embedded relations for multi-triple extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 659–665, Seattle, United States. Association for Computational Linguistics.

Benfeng Xu, Quan Wang, Yajuan Lyu, Yong Zhu, and Zhendong Mao. 2021. Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14149–14157.

Zhiheng Yan, Chong Zhang, Jinlan Fu, Qi Zhang, and Zhongyu Wei. 2021. A partition filter network for joint entity and relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 185–197, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hongbin Ye, Ningyu Zhang, Shumin Deng, Mosha Chen, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Contrastive triple extraction with generative transformer. In *AAAI*, pages 14257–14265. AAAI Press.

Xiaofeng Yu and Wai Lam. 2010. Jointly identifying entities and extracting relations in encyclopedia text via A graphical model approach. In *COLING 2010: Posters*, pages 1399–1407.

Yue Yuan, Xiaofei Zhou, Shirui Pan, Qiannan Zhu, Zeliang Song, and Li Guo. 2020. A Relation-Specific Attention Network for Joint Entity and Relation Extraction. In *IJCAI 2020*, pages 4054–4060.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106.

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 *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 367–377, Hong Kong, China. Association for Computational Linguistics.

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 *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514, Melbourne, Australia. Association for Computational Linguistics.

Hengyi Zheng, Rui Wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Xu Ming, and Yefeng Zheng. 2021a. PRGC: potential relation and global correspondence based joint relational triple extraction. In *ACL/IJCNLP 2021*, pages 6225–6235.

Hengyi Zheng, Rui Wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Xu Ming, and Yefeng Zheng. 2021b. PRGC: Potential relation and global correspondence based joint relational triple 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 6225–6235, Online. Association for Computational Linguistics.

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. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.

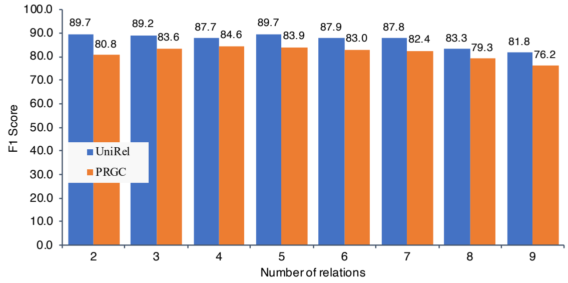


Figure 6: F1-score w.r.t. different number of relations.

A Supplemental Experiments

A.1 Influence of Random Seeds

As shown in Table 5 , we conduct experiments with 5 different random seeds, and the results and their improvements are rather robust. We get an averaged performance of 93.62±0.07/94.52±0.12 on NYT/WebNLG with 5 runs in total, which robustly outperforms the previous SOTA methods.

Dataset	2019	2021	2022	2023	2024
NYT	93.6	93.7	93.6	93.7	93.5
WebNLG	94.4	94.7	94.5	94.4	94.6

Table 5: F1-score of UniRel with different random seeds.

A.2 Influence of Relation Numbers

To study how the number of relations M influences the performance of UniRel, as shown in Figure 6, we conduct ablation experiments with different M. We control the number of sentences of each relation in the range of [1500, 1700) to keep each relation has similar signals in each experiment.

From Figure 6, we can observe that UniRel has relatively stable improvements when the number of relations increases. For example, UniRel achieves both +5.6% improvements when $M = 3$ and $M = 9$. The results show the effectiveness of UniRel is relatively stable with different numbers of relations.

A.3 Relation-Word Mapping

For clarity, we convert relations to natural language texts with human-picked words. The picked word is basically the most informative word within the label name that preserves its semantics, for example, “founders” for “/business/company/founders”. We don’t exploit any special selection strategy and need no formal annotation. For exceptional cases where the relation’s labels are very similar, we simply resort to alternative words ("part” and "section”) or capitalization ("part” and "Part”) to make a distinction. This provides a semantic-aware initialization and the model will continue to optimize them. As a result, the performance of UniRel is very robust to choices of words. As shown in Table 7, we conduct experiments with two different relation-word mappings of the NYT dataset, and the results show UniRel is robust to the mapping words.

A.4 Extended to Multi-token Entity Setting

With minimum adaptation, we can extend UniRel to the multi-token entity setting. The adaption processes are as follows: 1) repeat Interaction Map twice respectively for head and tail token of entity span to identify (subject-head, relation, object-head) and (subject-tail, relation, object-tail). 2) add a third Interaction Map between head-tail tokens to identify (head, relation, tail). 3) decode triples by linking the head and tail tokens of entities w.r.t. each relation.

As shown in Table 6, UniRel achieves SOTA performance under the multi-token entity setting, which further indicates the effectiveness of the proposed methods.

Model	NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1
NovelTagging	32.8	30.6	31.7	52.5	19.3	28.3
MultiHead	60.7	58.6	59.6	57.5	54.1	55.7
ETL-span	85.5	71.7	78.0	84.3	82.0	83.1
RSAN	85.7	83.6	84.6	80.5	83.8	82.1
PRGC$_{BERT}$	93.5	91.9	92.7	89.9	87.2	88.5
TPLinker$_{BERT}$	91.4	92.6	92.0	88.9	84.5	86.7
GRTE$_{BERT}$	93.4	**93.5**	**93.4**	**92.3**	87.9	90.0
OneRel$_{BERT}$	93.2	92.6	92.9	91.8	90.3	91.0
UniRel	**93.7**	93.2	**93.4**	91.8	**90.5**	**91.1**

Table 6: Results of the multi-token entity setting. The highest scores are in bold.

Relation	Mapping A	Mapping B
/business/company/advisors	advisors	counselor
/business/company/founders	founders	creator
/business/company/industry	industry	sector
/business/company/major_shareholders	holding	shareholder
/business/company/place_founded	founded	establish
/business/company_shareholder/major_shareholder_of	shareholder	holder
/business/person/company	company	corporation
/location/administrative_division/country	country	state
/location/country/administrative_divisions	administrative	administration
/location/country/capital	capital	Capital
/location/location/contains	contains	include
/location/neighborhood/neighborhood_of	neighbor	neighbour
/people/deceased_person/place_of_death	death	Death
/people/ethnicity/geographic_distribution	geographic	Geographic
/people/ethnicity/people	people	People
/people/person/children	children	Children
/people/person/ethnicity	ethnicity	ethnic
/people/person/nationality	nationality	national
/people/person/place_lived	lived	live
/people/person/place_of_birth	birthplace	birth
/people/person/profession	profession	career
/people/person/religion	religion	Religion
/sports/sports_team/location	location	Location
/sports/sports_team_location/teams	teams	Teams
Prec./Recall/F1.	93.5/94.0/93.7	93.9/93.4/93.6

Table 7: Two different relation-word mappings for NYT