TDEER: An Efficient Translating Decoding Schema for Joint Extraction of Entities and Relations

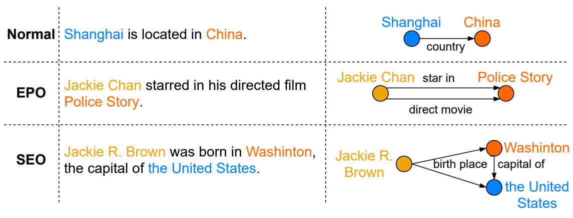
Xianming Li, Xiaotian Luo, Chenghao Dong [*], Daichuan Yang, Beidi Luan, Zhen He
Ant Group, Shanghai
`{niming.lxm,lxt267638,dch166451,beidi.lbd}@antgroup.com`
`{daichuan.ydc,fuyi.hz}@antgroup.com`

Abstract

Joint extraction of entities and relations from unstructured texts to form factual triples is a fundamental task of constructing a Knowledge Base (KB). A common method is to decode triples by predicting entity pairs to obtain the corresponding relation. However, it is still challenging to handle this task efficiently, especially for the overlapping triple problem. To address such a problem, this paper proposes a novel efficient entities and relations extraction model called **TDEER**, which stands for **T**ranslating **D**ecoding Schema for Joint **E**xtraction of **E**ntities and **R**elations. Unlike the common approaches, the proposed translating decoding schema regards the relation as a translating operation from subject to objects, i.e., TDEER decodes triples as $subject + relation \rightarrow objects$. TDEER can naturally handle the overlapping triple problem, because the translating decoding schema can recognize all possible triples, including overlapping and non-overlapping triples. To enhance model robustness, we introduce negative samples to alleviate error accumulation at different stages. Extensive experiments on public datasets demonstrate that TDEER produces competitive results compared with the state-of-the-art (SOTA) baselines. Furthermore, the computation complexity analysis indicates that TDEER is more efficient than powerful baselines. Especially, the proposed TDEER is 2 times faster than the recent SOTA models. The code is available at https://github.com/4AI/TDEER.

1 Introduction

Extraction of entities and relations from unstructured texts is one of the most essential information extraction tasks. It aims to extract entities and their corresponding semantic relations from unstructured texts, which are usually presented in a



Shanghai is located in China.

Jackie Chan starred in his directed film Police Story.

Jackie R. Brown was born in Washinton, the capital of the United States.

Figure 1: Cases of *Normal*, *EntityPairOverlap* (EPO), and *SingleEntityOverlap* (SEO) overlapping triples.

triple form of (*subject*, *relation*, *object*), e.g., (*Microsoft*, *co-founder*, *Bill Gates*). It is also a crucial step in building a large-scale KB and exerts an important role in the development of web search (Szumlanski and Gomez, 2010), question answering (Fader et al., 2014), biomedical text mining (Huang and Lu, 2016), etc.

Traditional approaches (Zelenko et al., 2003; Chan and Roth, 2011; Rink and Harabagiu, 2010) handle this task in a pipeline manner, i.e., extracting the entities first and then identifying their relations. The pipeline framework simplifies the extraction task, but it ignores the relevance between entity identification and relation prediction. To address this problem, several joint learning models have been proposed and can be categorized into feature-based models and end-to-end deep models. Feature-based models (Li and Ji, 2014; Ren et al., 2017) introduce a complex process of feature engineering and profoundly depend on Natural Language Processing tools for feature extraction. More recently, the end-to-end neural network models (Gupta et al., 2016; Zheng et al., 2017; Zeng et al., 2018; Fu et al., 2019; Wei et al., 2020) have become the mainstream method for relation extraction tasks. Such models utilize the learned representation from pre-trained language models and are a more promising approach than manual features.

More research interests have been concerned with complicated entity and relation extraction problems, such as the overlapping triple problem.

[*]Corresponding author.

Zeng et al. (2018) summarized the overlapping triple problem into three categories, i.e. *Normal*, *SEO*, and *EPO*, which are depicted in Figure 1. Many methods have been proposed to address the overlapping issue, for instance, encoder-decoder framework (Zeng et al., 2018) and decomposition approaches (Yu et al., 2020; Wei et al., 2020). However, such approaches still suffer from setbacks when handling the overlapping triple problem. More specifically, the encoder-decoder framework can only resolve the one-word entity overlapping problem and fail to handle the multi-word entity overlapping problem. Meanwhile, the decomposition approaches suffer error accumulation between dependent stages. To address these problem, Wang et al. (2020) presented a one-stage method, TPLinker, that transforms the joint extraction task into a token pair linking problem to resolve the overlapping triple problem. TPLinker does not contain any inter-dependent stages, hence it can alleviate error accumulation. However, processing all token pairs at encoder layers suffers from high computational complexity, which is an obstacle for TPLinker to encode long text.

We present a novel framework TDEER to jointly extract the entities and relations by a translating decoding schema to handle the overlapping triple problem. More concretely, TDEER interprets the relation as a translating operation from subject entity to object entities, i.e., it decodes triples by $subject + relation \rightarrow objects$. The proposed translating decoding schema can effectively resolve the overlapping triple problem. TDEER iterates all pairs of subjects and relations to recognize objects (or no object), hence all possible triples, including overlapping or non-overlapping triples, can be considered. We propose a negative sample strategy to detect and alleviate error propagation in different stages. This strategy can enable TDEER to alleviate error accumulation to achieve higher results. TDEER is an efficient approach as it first retrieves all possible relations and entities, then uses distinguished entities and relations to decode triples. By doing this, the search space can be reduced, thus it is more efficient than previous works. The computational complexity of the proposed translating decoding schema is $O(n + sr)$, where n is the sequence length, s is the number of subjects in the input sentence, r denotes the number of relations in the input sentence. Extensive experiments illustrate that TDEER achieves better results than SOTA models in most datasets and is competent in handling the overlapping triple problem.

In summary, our contributions are as follows: (1) We propose a novel translating decoding schema for joint extraction of entities and relations from unstructured texts. (2) TDEER can handle the intractable overlapping triple problem effectively and efficiently. (3) Notably, TDEER is about 2 times faster than the current SOTA models.

2 Related Work

The pipeline approach and joint approach are the two mainstream methods for extracting entities and relations from unstructured texts.

Traditionally, extracting entities and relations to form triples has been studied as two separated independent tasks: Named Entity Recognition (NER) and Relation Extraction. Mintz et al. (2009) introduced a distant supervision model, and Hoffmann et al. (2011) used a weak supervision method to extract entities and relations. The features of distant supervision and weak supervision approaches are often derived from Natural Language Processing (NLP) tools. It suffers from data labeling errors that inevitably exist in NLP tools. To address this problem, Zeng et al. (2015) employed a multi-instance learning approach to tackle the problem of data labeling errors. Qin et al. (2018) applied reinforcement learning for extraction of entities and relations. Although the pipeline models produced promising results, they neglect the triple-level dependencies between entities and relations. Recently, Zhong and Chen (2021) presented a pipeline approach incorporating entity information for entity and relation extraction.

To exploit the dependencies between entities and relations, multiply joint extraction models have been proposed. Zheng et al. (2017) introduced a unified tagging scheme and transformed the relationship extraction problem into a sequence labeling problem. Zeng et al. (2018) applied a sequence-to-sequence model with a copy mechanism to solve the overlapping triple problem. Trisedya et al. (2019) employed the encoder-decoder framework to jointly extract triples from sentences and map them into an existing KB. Fu et al. (2019) applied graph convolutional networks to jointly learn named entities and relations. Dai et al. (2019) presented a unified joint extraction model to tag entity and relation labels directly according to a query word position, which can simultaneously extract all entities and their types. Wei et al. (2020) proposed a cascade binary tagging framework. Wang et al. (2020) formulated the joint extraction as a token pair linking problem.

Moreover, some knowledge representation models (Bordes et al., 2013; Weston et al., 2013; Wang et al., 2014; Tu et al., 2017) are adopted to refine the triple extraction model via scoring candidate facts by knowledge graph embedding. Although some of them also use the "translation" idea, the function is different from ours. In their setting, they use the "translation" idea to construct rank-based knowledge graph embedding models. They cannot be used to extract entities and relations from texts directly. In our setting, the "translation" idea is applied to end-to-end joint extract entities and relations from text.

The proposed translating decoding schema is a novel approach to solve the overlapping triple problem effectively and efficiently, which makes our model crucially different from previous works.

3 Methodology

This paper proposes a three-stage model, TDEER. In the first stage, TDEER uses a span-based entity tagging model to extract all subjects and objects. In the second stage, TDEER employs the multi-label classification strategy to detect all relevant relations. In the third stage, TDEER iterates the pairs of subjects and relations to identify respective objects by the proposed translating decoding schema. Figure 2 shows the generic framework of TDEER. In subsequent sections, we will describe the three stages of TDEER in detail.

3.1 Input Layer

The input of TDEER is a sentence T. We pad the sentence to keep a uniform length n for all sentences. For an LSTM-based model, we first map each word into a k-dimensional continuous space and obtains the word embedding $\mathbf{t}_i \in \mathbb{R}^k$. Then we concatenate all word vectors to form a $k \times n$ matrix as model input: $\mathbf{t} = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_n]$. we employ LSTM on the embedding matrix to produce latent semantic feature map $\mathbf{X}$:

$$\mathbf{X} = \text{LSTM}(\mathbf{t}). \tag{1}$$

As for a BERT-based model, TDEER extracts feature map via the pre-trained BERT (Devlin et al., 2019) over text input:

$$\mathbf{X} = \text{BERT}(T). \tag{2}$$

3.2 Entity Tagging Model

To obtain entities and their positions efficiently, we adopt a span-based tagging model following prior works (Yu et al., 2020; Wei et al., 2020). We apply two binary classifiers to predict the start and end position of entities respectively. The operations on each token in a sentence are as follows:

$$p_i^{start} = \sigma(\mathbf{W}_{start}\mathbf{X}_i + \mathbf{b}_{start}), \tag{3}$$

$$p_i^{end} = \sigma(\mathbf{W}_{end}\mathbf{X}_i + \mathbf{b}_{end}), \tag{4}$$

where p_i^{start} and p_i^{end} stand for the probabilities of recognizing the i-th token in input sequence as the start and end position of an entity, respectively. $\sigma(\cdot)$ denotes a sigmoid activation function.

The entity tagging model is trained by minimizing the following loss function:

$$\mathcal{L}^e = -\frac{1}{n}(log\ p_\theta^{start}(s|\mathbf{X}) + log\ p_\theta^{end}(s|\mathbf{X})), \tag{5}$$

where $p_\theta(s|\mathbf{X}) = \prod_{i=1}^{n} p_i^{\mathbb{I}\{y_i=1\}}(1-p_i)^{\mathbb{I}\{y_i=0\}}$, p_θ^{start} is the likelihood for the start positions, and p_θ^{end} is the likelihood for the end positions.

We apply the entity tagging model to obtain all subjects and objects in one sentence. Detected subjects and objects are denoted as $\mathbf{\Omega}_s$ and $\mathbf{\Omega}_o$, respectively. The extracted entity is presented into a tuple like (*start*, *end*).

3.3 Relation Detector

In general, more than one relation can be detected in a sentence. For example, there are four relations *Star In*, *Direct Movie*, *Live In*, and *Capital Of* in the sentence in Figure 2. To identify related relations in a sentence, we adopt a multi-label classification strategy. For the BERT-based/LSTM-based model, we project the "[CLS]" token/last output (LO) representation into a relation-detection space for multi-label classification, as follows:

$$\mathbf{r} = \sigma(\mathbf{W}_{rel}\mathbf{X}_{CLS/LO} + \mathbf{b}_{rel}), \tag{6}$$

where $\sigma(\cdot)$ denotes sigmoid function.

The relation detector minimizes the following binary cross-entropy loss function to detect relations:

$$\mathcal{L}^r = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i\log(r_i) + (1-y_i)\log(1-r_i)\right], \tag{7}$$

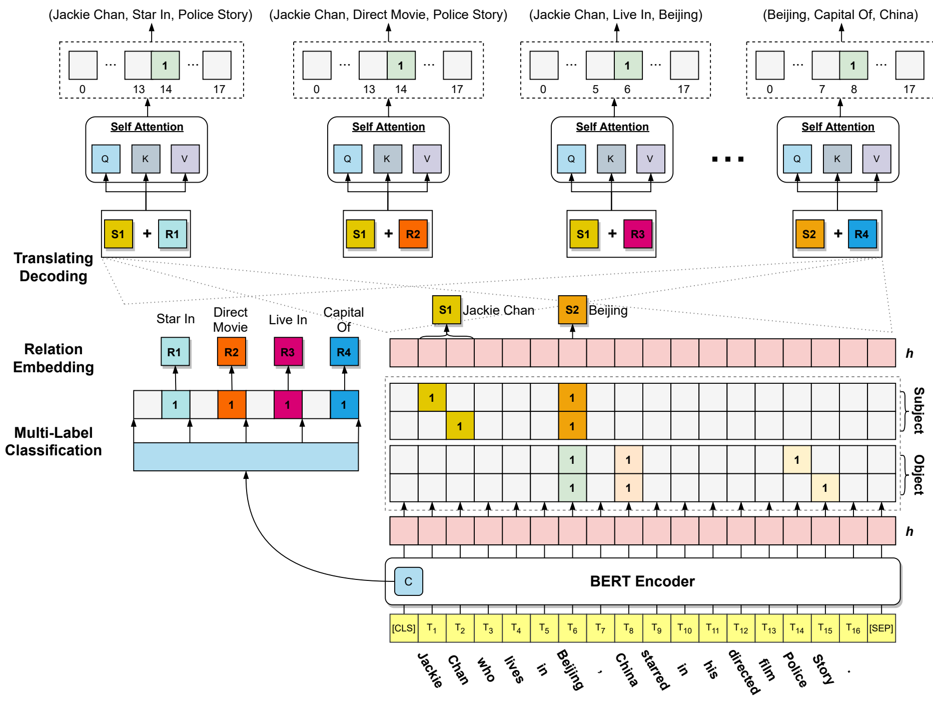


Figure 2: The generic framework of TDEER. In this example, TDEER first identifies *Jackie Chan* and *Beijing* as subjects and recognizes *Beijing*, *China*, and *Police Story* as objects. Then TDEER detects four involved relations: *Star In*, *Direct Movie*, *Live In*, and *Capital Of*. Finally, TDEER decodes triples by iterating all pairs of subjects and relations to recognize objects: (*Jackie Chan*, *Star In*) $\rightarrow$ *Police Story*, (*Jackie Chan*, *Direct Movie*) $\rightarrow$ *Police Story* (So far, the EPO problem can be resolved.), (*Jackie Chan*, *Live In*) $\rightarrow$ (*Beijing*, *China*), ..., (*Beijing*, *Capital Of*) $\rightarrow$ *China* (by now, the SEO problem can be resolved).

where $y_i \in \{0, 1\}$ indicates the ground truth label of relations. We denote the detected relations in a sentence as $\mathbf{\Omega}_r$.

3.4 Translating Decoding Schema

We iterate the pairs of detected subjects $\mathbf{\Omega}_s$ and relations $\mathbf{\Omega}_r$ to predict the start positions of objects. For each subject and relation pair, we first combine the representation of subject and relation. Next, we use the attention mechanism to obtain a selective representation, which is expected to assign higher weights to possible positions of objects. Finally, we pass the selective representation to a fully-connect layer to get the output, i.e. the positions of objects.

More concretely, for the i-th subject in $\mathbf{\Omega}_s$ and j-th relation in $\mathbf{\Omega}_r$, TDEER takes the averaged vector span representation between the start and end tokens of the subject as $\mathbf{v}^i_{sub}$. TDEER maps the relation into a continuous space with the same feature dimension as $\mathbf{v}^i_{sub}$ to produce the relation embedding vector $\mathbf{e}^j_{rel}$. Then TDEER applies a fully-connect layer to encode the relation:

$$\mathbf{v}^j_{rel} = \text{FullyConnect}(\mathbf{e}^j_{rel}). \tag{8}$$

TDEER links subject and relation via addition operation, as follows:

$$\mathbf{u} = \mathbf{v}^i_{sub} + \mathbf{v}^j_{rel}. \tag{9}$$

We adopt the addition operation because it is intuitive, and it does not change the tensor shape of inputs, which is convenient for attention computation.

Next, TDEER applies the attention mechanism to obtain the selective representation.

$$\begin{aligned}\mathbf{A} &= \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}\\ \mathbf{Q} &= \mathbf{W}_{query}(\mathbf{X}+\mathbf{u}) + \mathbf{b}_{query}\\ \mathbf{K} &= \mathbf{W}_{key}(\mathbf{X}+\mathbf{u}) + \mathbf{b}_{key}\\ \mathbf{V} &= \mathbf{W}_{value}(\mathbf{X}+\mathbf{u}) + \mathbf{b}_{value},\end{aligned} \tag{10}$$

where d_k is the dimension of the attention key. Furthermore, TDEER adopts a binary classifier to identify the start positions of objects given the current subject and relation.

$$p^{obj\_start}_i = \sigma(\mathbf{W}_{trans}\mathbf{A}_i + \mathbf{b}_{trans}), \tag{11}$$

where $p^{obj\_start}_i$ indicates the probability of identifying the i-th token in input sequence as the start position of an object entity.

In this stage, TDEER minimizes the following loss function to discern the start positions of object entities.

$$\mathcal{L}^t = -\frac{1}{n}\sum_{i=1}^{n}\Big[\mathbb{I}\{y_i = 1\}\log(p_i^{obj\_start}) + \mathbb{I}\{y_i = 0\}\log(1 - p_i^{obj\_start})\Big], \tag{12}$$

where $\mathbb{I}$ is the indicator function. After obtaining the start positions of objects, TDEER takes the corresponding entities from the $\mathbf{\Omega}_o$ which has the same start position as the final objects. If no start positions match, there is no triple for the current subject and relation.

3.5 Negative Sample Strategy

Most entities and relations extraction models consisting of multiple components suffer from error accumulation. Errors from upstream components will propagate to downstream components because of the dependency between components. In TDEER, the translating decoder is dependent on the entity tagger and relation detector, hence the translating detector may receive error entities or relations from upstream components. Therefore, we introduce a negative sample strategy to detect and alleviate errors from upstream components. For each sentence, we produce incorrect triples as negative samples by replacing the correct subject/relation with other inappropriate subjects/relations during the training phase. We do not assign any objects to negative samples, namely the probabilities of start positions of Eq.(11) are all expected to be 0. This strategy enables TDEER to handle noisy inputs of subjects and relations at the decoding phase.

3.6 Joint Training

We jointly train the span-based entity tagging model, the relation detector, and the translating decoder. The joint loss function is defined as follows:

$$\mathcal{L} = \alpha\mathcal{L}^e + \beta\mathcal{L}^r + \lambda\mathcal{L}^t, \tag{13}$$

where α, β and λ are constants. In our experiment, we set 1.0, 1.0, and 5.0, respectively. The values are obtained by grid search on the validation set.

4 Experiment

4.1 Datasets and Evaluation Metrics

We conduct experiments on widely used datasets. **NYT** (Riedel et al., 2010) dataset was produced by distant supervision method from New York Times news articles. **WebNLG** was created for Natural Language Generation and adapted by Zeng et al. (2018) for relational triple extraction. For a fair comparison, we apply the two datasets released by Zeng et al. (2018). Apart from evaluating the model on standard splitting, we follow (Wei et al., 2020) to partition the test sentences according to different overlapping categories, different triple numbers for experiments on overlapping triples, and various triple numbers. Furthermore, we also conduct experiments on **NYT11-HRL** (Takanobu et al., 2019), in which most test sentences belong to *Normal*, to demonstrate that the proposed model can handle not only the overlapping triple problem but also the general problem. The adopted public datasets with summary statistics in Table 1.

We report the standard micro Precision, Recall, and F1-score following the same setting in Fu et al. (2019).

4.2 Baselines

We compare the proposed model with following SOTA models: **NovelTagging** (Zheng et al., 2017) incorporates both entity and relation roles and models relational triple extraction problem as a sequence labeling problem; **CopyR** (Zeng et al., 2018) applies a sequence-to-sentence architecture; **GraphRel** (Fu et al., 2019) uses graph convolutional networks to jointly learn named entities and relations; **OrderCopyR** (Zeng et al., 2019) applies the reinforcement learning into an sequence-to-sequence model to generate triplets; **CasRel** (Wei et al., 2020) employs a cascade binary tagging framework; **TPLinker** (Wang et al., 2020) iterates all token pairs and use matrices to tag token links to recognize relations between token pair.

4.3 Implementation Details

We adopt the Adam (Kingma and Ba, 2015) optimizer. The hyper-parameters are tuned by grid search on the validation set. The learning rate is set to 1e-3/5e-5 and the batch size is set to 32/8 in the backbone as LSTM/BERT. For the LSTM-based model, we apply the 300-dimension pre-trained GloVe embedding (Pennington et al., 2014) [1]. For the BERT-based model, we use the pre-trained Cased BERT-base [2] as the backbone.

[1]http://nlp.stanford.edu/data/glove.6B.zip

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

Category	Train	Valid	Test	Overlapping Pattern			Number of triples					Rel.
				Norm.	SEO	EPO	$= 1$	$= 2$	$= 3$	$= 4$	≥ 5	
NYT	56195	5000	5000	3266	1297	978	3244	1045	312	291	108	24
WebNLG	5019	500	703	246	457	26	266	171	131	90	45	216*
NYT11.	62648	–	369	368	1	0	368	1	0	0	0	12

Table 1: Statistics of datasets. NYT11. denotes NYT11-HRL, Norm. stands Normal triples, and Rel. stands Relations. Note that the relation number of WebNLG was miswritten as 246 in (Zeng et al., 2018; Wei et al., 2020). We correct the number in the statistics and use $\star$ to mark the accurate number.

4.4 Results & Discussion

4.4.1 Main Results

The main results of the proposed models and baseline models are reported in Table 2. CasRel, TPLinker, and TDEER achieve absolute improvements on NYT and WebNLG datasets against the rest baselines. Especially, TDEER produces competitive results compared with the previous SOTA model TPLinker and achieves 7 out of 9 best results. Moreover, TDEER outperforms baseline models over the F1 score on all datasets. From Table 1, we can observe that the data size of WebNLG is small while it consists of a large number of predefined relations. It is difficult to make improvements on WebNLG, as existing models can achieve an F1 score over 90%, which has already exceeded human-level performance (Wang et al., 2020). Even though, TDEER achieves around 1.2% gain to 93.1% on WebNLG against TPLinker, which verifies the effectiveness of the proposed framework. Apart from F1, we find that TDEER performs better on precision score than baselines models in most results.

Although without a pre-trained language model as the backbone, $\text{TDEER}_{\text{LSTM}}$ still performs well. $\text{TDEER}_{\text{LSTM}}$ achieves a higher F1 score on NYT against baseline models except for BERT-Based CasRel and TPLinker. Furthermore, $\text{TDEER}_{\text{LSTM}}$ outperforms baseline models on WebNLG and NYT11-HRL over precision against all baseline models except for BERT-Based CasRel. Therefore, the proposed framework is efficacious even though without a powerful pre-trained language model.

NYT and WebNLG contain a large number of overlapping-triple instances. Therefore, the results on NYT and WebNLG indicate that TDEER can address the overlapping triple problem. Almost all triples in NYT11-HRL belong to *Normal*. TDEER achieves better results than baselines on NYT11-HRL, which shows that TDEER can solve the general extraction problem.

4.4.2 Results of Ablation Study

We conduct ablation studies on different strategies to explore the effect of the negative sample strategy. It shows that negative samples from subjects achieve better results than negative samples from relations. TDEER performs better by combining the two types of negative sample strategy than adopting each strategy individually or without negative samples. This evidence illustrates that negative samples are helpful to alleviate error accumulation. We also conduct ablation studies on TDEER without the relation detector or attention. It also shows that the results of TDEER are better than TDEER without the relation detector or attention. We notice that the model will malfunction without the relation detector. This evidence suggests that the relation detector and attention are crucial for TDEER.

To investigate the effect of the attention mechanism, we pick up a sample from the NYT test set which contains a triple (*Netherlands*, */location/country/administrative_division*, *Utrecht*). We visualize the attention heatmap of different subject and relation pairs as depicted in Figure 3. The heatmap indicates that when the extracted subject and relation pair are proper, the attention weights on object positions are higher than others. If the weights are close to each other, then the extracted subject and relation pair can not be decoded to form a valid triple.

4.4.3 Discussion on Triple Numbers

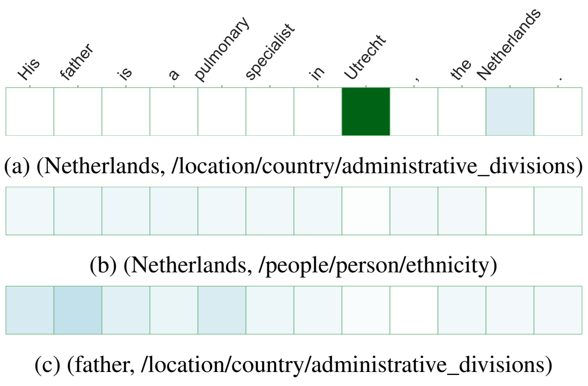
In general, the more triples in a sentence, the more complicated the sentence is. To explore the model performance regarding different sentence complexities, we also conduct experiments on sentences with different triple numbers. The results are reported in Table 4. From the results, we can find that TDEER outperforms baseline models except for four triple numbers in NYT. Notably, TDEER

Model	NYT			WebNLG			NYT11-HRL		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
NovelTagging	62.4	31.7	42.0	52.5	19.3	28.3	46.9	48.9	47.9
CopyR	61.0	56.6	58.7	37.7	36.4	37.1	34.7	53.4	42.1
OrderCopyR	77.9	67.2	72.1	63.3	59.9	61.6	–	–	–
GraphRel	63.9	60.0	61.9	44.7	41.1	42.9	–	–	–
$\text{CasRel}_{\text{LSTM}}$	84.2	83.0	83.6	86.9	80.6	83.7	–	–	–
$\text{CasRel}_{\text{BERT}}$	89.7	89.5	89.6	93.4	90.1	91.8	50.1	**58.4**	53.9
$\text{TPLinker}_{\text{LSTM}}$	83.8	83.4	83.6	90.8	90.3	90.5	–	–	–
$\text{TPLinker}_{\text{BERT}}$	91.3	**92.5**	91.9	91.8	92.0	91.9	–	–	–
$\text{TDEER}_{\text{LSTM}}$	85.5	82.4	83.9	92.1	86.9	89.4	57.7	43.3	49.5
$\text{TDEER}_{\text{BERT}}$	**93.0**	92.1	**92.5**	**93.8**	**92.4**	**93.1**	**63.5**	55.7	**59.3**

Table 2: Results of baseline models on NYT, and WebNLG datasets. The results of baselines are retrieved from original papers respectively.

Model	F1	
	NYT	WebNLG
w/o neg.	69.5	67.7
w/o neg sub.	81.1	86.1
w/o neg rel.	82.3	86.8
w/o rel.	13.5	2.7
w/o attn.	82.7	87.3
$\text{TDEER}_{\text{LSTM}}$	**83.9**	**89.4**

Table 3: Results of ablation study. w/o neg. stands for $\text{TDEER}_{\text{LSTM}}$ without negative samples. w/o neg sub. stands for $\text{TDEER}_{\text{LSTM}}$ without negative samples from subjects. w/o neg rel. denotes $\text{TDEER}_{\text{LSTM}}$ without negative samples from relations. w/o rel. denotes $\text{TDEER}_{\text{LSTM}}$ without relation detector. w/o attn. means $\text{TDEER}_{\text{LSTM}}$ without attention.



(a) (Netherlands, /location/country/administrative_divisions)

(b) (Netherlands, /people/person/ethnicity)

(c) (father, /location/country/administrative_divisions)

Figure 3: Attention heatmap of different pairs of subject and relation. The attention weight is higher when the color is deeper.

achieves 2.8% gain in NYT and 0.7% gain in WebNLG against TPLinker for complicated sentences containing five or more triples. This evidence illuminates that TDEER is effective to model sentences with multiple triples.

4.4.4 Discussion on Overlapping Patterns

To further investigate the performance of different overlapping patterns, we conducted extensive experiments and report the results on different overlapping patterns on NYT and WebNLG in Table 5. The results suggest that TDEER outperforms baseline models, which demonstrates the advantages of TDEER in processing the overlapping triple problem.

4.4.5 Discussion on Computation Complexity

Computation efficiency is an important problem that is not paid enough attention to by most previous works. We compare TDEER with baselines in computational complexity and inference time on the test set. The results are shown in Table 6.

Pipeline approaches usually use NER tools to detect entities. NER tools usually apply the Viterbi algorithm to decode sequences with $O(nK^2)$ complexity, where n denotes the input length and K is the tag size. Pipeline approaches recognize relations from each entity pair. Thus, the computation complexity of pipeline approaches is $O(nK^2 + e^2)$, where e denotes the number of entities in the input.

Despite the successes of CasRel (Wei et al., 2020) and TPLinker (Wang et al., 2020), they still struggle with computation efficiency. CasRel jointly decodes relations and objects. The com-

Model	NYT					WebNLG				
	N=1	N=2	N=3	N=4	N $\geq$ 5	N=1	N=2	N=3	N=4	N $\geq$ 5
CopyR†	67.1	58.6	52.0	53.6	30.0	59.2	42.5	31.7	24.2	30.0
GraphRel†	71.0	61.5	57.4	55.1	41.1	66.0	48.3	37.0	32.1	32.1
OrderCopyR†	71.7	72.6	72.5	77.9	45.9	63.4	62.2	64.4	57.2	55.7
CasRel$_{\text{BERT}}$	88.2	90.3	91.9	94.2	83.7	89.3	90.8	94.2	92.4	90.9
TPLinker$_{\text{BERT}}$	90.0	**92.8**	93.1	**96.1**	90.0	88.0	90.1	**94.6**	93.3	91.6
TDEER$_{\text{BERT}}$	**90.8**	**92.8**	**94.1**	95.9	**92.8**	**90.5**	**93.2**	**94.6**	**93.8**	**92.3**

Table 4: F1-score on sentences with different triple numbers. † stands results retrieve from (Wei et al., 2020). The results of the rest of the baselines are obtained from the respective original paper.

Model	NYT			WebNLG		
	Normal	SEO	EPO	Normal	SEO	EPO
CopyR	66.0	48.6	55.0	59.2	33.0	36.6
OrderCopyR	71.2	69.4	72.8	65.4	60.1	67.4
GraphRel	69.6	51.2	58.2	65.8	38.3	40.6
CasRel$_{\text{BERT}}$	87.3	91.4	92.0	89.4	92.2	94.7
TPLinker$_{\text{BERT}}$	90.1	93.4	94.0	87.9	92.5	95.3
TDEER$_{\text{BERT}}$	**90.8**	**94.1**	**94.5**	**90.7**	**93.5**	**95.4**

Table 5: Results of models in *Normal*, *EntityPairOverlap* and *SingleEntityOverlap* patterns in NYT and WebNLG datasets. Results of baselines are obtained from respective original paper.

Model	Complexity	Time (ms)	
		NYT	WebNLG
Pipeline	$O(nK^2 + e^2)$	–	–
CasRel	$O(n + sro)$	54.0*	76.8*
TPLinker	$O(n^2)$	82.7*	112.6*
TDEER	$O(n + sr)$	**31.4**	**36.9**

Table 6: Computation efficiency. $\star$ denotes results retrieve from (Wang et al., 2020). e denotes the number of entities in input, *s*/*o* stands for the number of subjects/objects in input, r denotes the number of relations in input, K denotes the tag size, and n stands input length. Inference time presents the average time BERT-based models take to process a sample.

putational complexity of CasRel is $O(n + sro)$, where n is the input length, *s*/*r*/*o* is the number of subjects/relations/objects in the input, respectively. TPLinker iterates all token pairs and uses matrices to tag token links to recognize relations. The main computation overhead is on the encoder with $O(n^2)$ complexity, where n is the input length.

The computation complexity of TDEER is $O(n + sr)$, where n is the input length, s denotes the number of subjects in input, and r is the number of relations in the input sentence. It is 0.7 times faster than CasRel and 1.6 times faster than TPLinker on NYT, and 1.1 times faster than CasRel and 2.1 times faster than TPLinker on WebNLG from Table 6. Therefore, we can conclude that TDEER is more efficient than baselines, which makes TDEER competent in constructing a large-scale KB.

5 Conclusion & Future work

In this paper, we have proposed a novel translating decoding schema for joint extraction of entities and relations, namely TDEER. It models the relation as a translating operation from subjects to objects, which can handle the overlapping triple problem naturally. We have conducted extensive experiments on widely used datasets to demonstrate the effectiveness and efficiency of the proposed model. The proposed negative sample strategy is used to alleviate the error accumulation problem. Though it is effective, it may increase training time. For future work, we plan to explore more efficient approaches to alleviate error accumulation.

References

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pages 2787–2795.

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.

Dai Dai, Xinyan Xiao, Yajuan Lyu, Shan Dou, Qiaoqiao She, and Haifeng Wang. 2019. Joint extraction of entities and overlapping relations using position-attentive sequence labeling. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pages 6300–6308.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1156–1165.

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.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 2537–2547.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550.

Chung-Chi Huang and Zhiyong Lu. 2016. Community challenges in biomedical text mining over 10 years: success, failure and the future. *Briefings Bioinform.*, 17(1):132–144.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

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*, pages 402–412.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1011.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018. Robust distant supervision relation extraction via deep reinforcement learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 2137–2147.

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 *Proceedings of the 26th International Conference on World Wide Web*, pages 1015–1024.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163.

Bryan Rink and Sanda M. Harabagiu. 2010. UTD: classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010*, pages 256–259.

Sean R. Szumlanski and Fernando Gomez. 2010. Automatically acquiring a semantic network of related concepts. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management*, pages 19–28.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pages 7072–7079.

Bayu Distiawan Trisedya, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. 2019. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 229–240.

Cunchao Tu, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2017. Transnet: Translation-based network representation learning for social relation extraction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 2864–2870.

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 *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1572–1582.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119.

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.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1366–1371.

Bowen Yu, Zhenyu Zhang, Xiaobo Shu, Tingwen Liu, Yubin Wang, Bin Wang, and Sujian Li. 2020. Joint extraction of entities and relations based on a novel decomposition strategy. In *Proceedings of the 24th European Conference on Artificial Intelligence*, pages 2282–2289.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research*, 3:1083–1106.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762.

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*, pages 367–377.

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*, pages 506–514.

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*, pages 1227–1236.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*.