IPED: An Implicit Perspective for Relational Triple Extraction based on Diffusion Model

Anonymous ACL submission

Abstract

Relational triple extraction is a fundamental task in the field of information extraction, and a promising framework based on table filling has recently gained attention as a potential baseline for entity relation extraction. However, inherent shortcomings such as redundant information and incomplete triple recognition remain problematic. To address these challenges, we propose an **I**mplicit **P**erspective for relational triple **E**xtraction based on **D**iffusion model (IPED), an innovative approach for extracting relational triples. Our classifier-free solution adopts an implicit strategy using block coverage to complete the tables, avoiding the limitations of explicit tagging methods. Additionally, we introduce a generative model structure, the block-denoising diffusion model, to collaborate with our implicit perspective and effectively circumvent redundant information disruptions. Experimental results on two popular datasets demonstrate that IPED achieves state-of-the-art performance while gaining superior inference speed and low computational complexity. To support future research, we have made our source code publicly available online. [1]

1 Introduction

The extraction of relational triples has been an important and fundamental task in knowledge graph construction (Zamini et al., 2022; Wei et al., 2020a), aiming to recognize triples in the form of (*head entity*, *relation*, *tail entity*) from unstructured text. Current research in information extraction can be categorized into two main approaches: the joint extraction models, which utilize a simultaneous style, and the pipeline models, which utilize a two-encoder methodology to extract entities and relations. While the pipeline framework is criticized for serious error propagation and lack of interaction between its two subtasks (Shen et al., 2021), leading to performance decline, many recent joint extraction models have begun to thrive due to their enhanced capability to deal with complex scenarios such as single entity overlap (SEO), entity pair overlap (EPO), and subject object overlap (SOO).

Among these popular joint extraction methods, one baseline, known as the table-filling method, has gained favor in recent research. Compared to a multi-task joint structure, this method features a table of token pair units that are to be filled and decoded in a single step. In this way, it avoids exposure bias and error propagation, challenges that most methods cannot fully overcome. Particularly for recently proposed models (Shang et al., 2022; Ren et al., 2021; Wang et al., 2021), these can employ a novel table-filling strategy to simplify the decoding process and enhance information interaction.

Despite many unique advantages over table-filling methods, some flaws still remain to be addressed. (1) The abundance of negative tagging in a table, which is significantly denser than positive tagging, leads to imbalanced labeling and redundant information (Wang et al., 2021; Ning et al., 2023). To the best of our knowledge, this is a universal issue across all table-filling models. This imbalance results in a bias towards negative tagging and heightened computational complexity. (2) Many table-filling strategies fail to extract all scenarios of triples, leading to decreased recall (Ning et al., 2023). Even in the recent significant work (Shang et al., 2022), entities consisting of a single token in a triple cannot be properly extracted due to conflicts arising from multiple labels in one element. (3) Once a sentence contains multiple triples, the separate labels of different triples may intersect in a single element, causing confusion in decoding all ground-truth triples. Many models (Ren et al., 2021; Ning et al., 2023) employ decoding algorithms that match labels based on the nearest-neighbor principle, which can lead to error

[1] Our source code will be released on GitHub.

associations within a triple. (4) A line of models, not limited to table-filling ones, exhibit poor learning performance on the WebNLG dataset in contrast to the NYT dataset and they attribute it to the vast number of predefined relations in the former dataset (Gao et al., 2023).

After conducting a detailed observation and analysis of their models, it is observed that all existing table-filling-based methods are consistently constrained by the approach of utilizing a classifier to tag each table element explicitly. Mainly because of this, most of them can hardly escape the challenges mentioned above, despite attempts to introduce innovative labeling strategies and creative decoding algorithms. This constraint necessitates traversing each element of the table, consequently leading to a substantial number of negative samplings. This explicit way of assigning a fixed label to each element can not cope with scenarios when one element requires multiple labels, leading to the inability to recognize all triples and confusion in the regions where triple labels intersect. Additionally, certain decoding strategies, designed in response to this approach, often result in incorrectly matched labels for a triple.

To address the aforementioned issues at a fundamental level, instead of explicitly labeling all the elements, we formulate a fresh perspective to implicitly fill the tables using a block-covered approach. In this method, blocks defined by four edges (up, down, right, left) and one level are refined within a three-dimensional table (multiple two-dimensional tables stacked together). In alignment with this implicit approach, we introduce a generative model designed to recover all blocks within the tables. Specifically, our proposed block-denoising diffusion model (Blk-DDM) can progressively refine the edges and levels of the initialized blocks step by step through a reverse process, ensuring the blocks precisely cover the ground truth triples horizontally, vertically, and deeply. As a result, our model naturally disregards redundant information by leaving the negative spaces alone rather than classifying them. Furthermore, our approach allows for the adequate recognition of all potential triples, as the proposed blocks can overlap implicitly. In contrast to previous decoding algorithms that match explicit labels, our proposed simple but effective Parallel Boundary Emitting Strategy (PBES) for decoding has the capability of extracting all triples accurately, circumventing error association challenges and significantly accelerating inference. Additionally, our denoising diffusion process enables the gradual refinement of specific fine-grained relation types within triples, enhancing performance in large-relation datasets such as WebNLG (demonstrated in the Ablation Study detailed in Appendix A). Experimental results on two datasets, NYT and WebNLG, demonstrate that our model achieves state-of-the-art performance and exhibits superior efficiency in inference.

2 Related Works

2.1 Joint Extraction Models

Existing joint extraction models can be roughly sorted into two frameworks. The first framework, based on multi-task learning, utilizes a shared encoder but employs distinct decoders to sequentially predict entities and relations. (Miwa and Bansal, 2016) proposes an integrated model that extracts entities and relations separately, leveraging shared parameters and mutual interaction. (Luan et al., 2018) adopts a model employing shared data representations to mitigate error propagation between tasks. CasRel (Wei et al., 2020b) treats relations as functions mapping subjects to objects to make extraction. The other framework is structured prediction which integrates the two subtasks into a unified structure and performs decoding in one step. (Katiyar and Cardie, 2017) proposes a model using sequence tagging-based approaches and forbidding dependency trees. (Sun et al., 2019) employs graph convolutional networks for joint inference. (Wang and Lu, 2020) implements a table-filling strategy using a table encoder and a sequence encoder.

2.2 Diffusion Model

Diffusion model is a type of deep latent generative model primarily utilized for generating continuous data structures, such as images and audio. DDPM (Ho et al., 2020) is a pioneer work that makes diffusion model practical to applications, thus inviting excellent works (Kong et al., 2021; Zhao et al., 2023) in various fields. Recently, there has been an emergence of works in NLP utilizing diffusion models, such as (Li et al., 2022a; He et al., 2023) in language model and (Bi et al., 2023; Gong et al., 2023) in sequence-to-sequence tasks, despite the perceived challenges in applying diffusion models to discrete text sequences. Notably, DiffusionNER (Shen et al., 2023) also applies the diffusion model to named entity recognition. However, there are significant differences with our IPED, particularly

in (1) task definition: IPED concentrates on extracting relational triples rather than mere entities. (2) core design: our model operates by diffusing in a three-dimensional space for each triple, in contrast to DiffusionNER, which diffuses within a one-dimensional matrix for each entity and incorporates an additional classifier.

3 Methodology

This section firstly introduces our implicit table-filling strategy and its corresponding decoding algorithm. Secondly, the formulation of the Block-Denoising Diffusion Model is presented. Finally, the network architecture of our model is detailed.

3.1 Implicit Block-Covered Table Filling

For a sentence $\mathcal{S} = \{x_1, x_2, ..., x_L\}$ composed of L words, K relations $\mathcal{R} = \{r_1, r_2, ..., r_K\}$ are predefined in a dataset. The objective of relational triple extraction is to identify all triples (*head*, *relation*, *tail*) in each sentence, where the head and tail represent the subject and object entities, respectively, along with their connected relation. Within a sentence, for all triples $\tau = \{(h_i, r_i, t_i)\}_{i=1}^{M}$, M denotes the total number of triples, and h_i, t_i represent the entity spans, each composed of one or more consecutive tokens.

Unlike previous classifier-based tagging methods, our model does not allocate a label to each unit of the L*L*K three-dimensional matrix. Instead, it refines M blocks ($\mathbf{B} \in \mathbb{R}^{M\times 5}$) to cover the K tables horizontally, vertically, and deeply, which is, our implicit way to fill the tables. As illustrated in Figure 1, each block consists of five elements: the up and down edges indicate vertical positioning, the left and right edges denote horizontal positioning, and the level represents depth positioning within the K stacked tables, with each table corresponding to a specific relation. Via our proposed Blk-DDM (described in Section 3.2), these M blocks are progressively refined to reveal the recognized triples.

The proposed decoding scheme, named Parallel Boundary Emitting Strategy (PBES), is introduced to extract triples from the blocks. PBES follows the four edges and one level of each block, emitting them in parallel to the corresponding entities and relation. Specifically, for each block, the up and down edges are extended to the left side of the table, indicating the boundaries of the head entity. Similarly, the left and right edges are extended correspondingly to identify the boundaries of the tail entity. Meanwhile, the depth level where the block is located signifies a specific table, thereby indicating a particular relation. By repeating this process M times as described, all blocks are converted into relational triples.

Our table-filling method enables the precise extraction of all existing triples by circumventing the conflicts typically associated with explicit tagging. Thanks to the lack of inner constraints between the M blocks, this approach not only naturally tackles complex scenarios such as SEO, EPO, and SOO, but also overcomes issues like the failure of single-token entity extraction in (Shang et al., 2022) and error association in (Ren et al., 2021; Ning et al., 2023).

3.2 Block-Denoising Diffusion Model

In this section, we present the formulation of block generation as a denoising diffusion process and introduce our block-denoising diffusion model (Blk-DDM). As depicted in Figure 1, the diffusion model comprises a forward process that incrementally introduces noise to data samples and a reverse process that recovers the ground truth through step-by-step denoising. These two processes are synchronized to facilitate the learning of a network endowed with the denoising capability. During the inference phase, the diffusion model incrementally refines data samples through a multistep denoising process from a standard Gaussian distribution. Consequently, we convert our M blocks, composed of five elements (up, down, left, right, level), into index format $\mathbf{B} = \{(u_i, d_i, l_i, r_i, v_i)\}_{i=0}^{M}$ to support the denoising operations. Following (Ho et al., 2020), the forward denoising process is simplified by computing $\{\bar{\alpha}_1, ..., \bar{\alpha}_T\}$ from a predefined variance schedule $\{\beta_t\}_{t=0}^{T} \in (0, 1)$, and thus noise injection in multiple steps can be integrated into one step as follows:

$$q\left(z_t \mid z_0\right) = \mathcal{N}\left(z_t; \sqrt{\bar{\alpha}_t} z_0, \left(1 - \bar{\alpha}_t\right) \mathbf{I}\right) \qquad (1)$$

where q represents the forward process from z_0 to z_t. z_0 and z_t denote the original data and the noised data at timestep t, respectively. $\mathbf{I}$ is the standard Gaussian distribution. Note that the fixed forward process depicted in Figure 1 can be considered as a Markov chain.

Training Process The training process of the diffusion model involves a one-step noise addition and a one-step prediction towards the ground

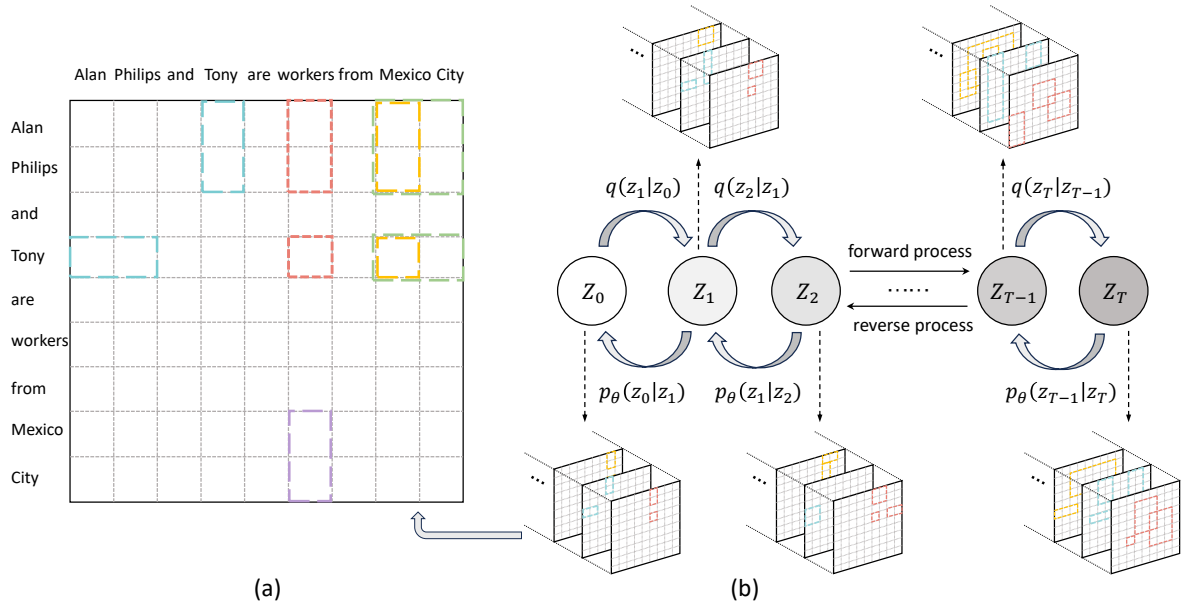

Figure 1: Figure (a) depicts our table-filling strategy. For the convenience of illustration, we simplify our three-dimensional tables into the form of a two-dimensional table in Figure (a), containing nine blocks in total that represent different triples. Here, dashed rectangles denote the four edges of the blocks, and different colors indicate the levels of the blocks. Figure (b) illustrates the overall diffusion process.

truth, aimed at training a network for inference purposes. As for a sentence, blocks $\mathbf{B} \in \mathbb{R}^{M\times 5}$ are initially derived from M ground truth triples. Subsequently, $\mathbf{B}$ is expanded by some blocks randomly sampled from a Gaussian distribution, resulting in $z_0 = \mathbf{B} \in \mathbb{R}^{N\times 5}$ (N > M). Following Equation (1), we then have

$$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{2}$$

where t ($\leq$ predefined total timestep T) is a randomly chosen timestep and $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ donates the pure noise from the Gaussian distribution, thus getting noised blocks $\mathbf{B}$. Feeding z_t into our network f_θ, one can get the predicted z_0 (Section 3.3) and compute the objective function (Section 3.3.3). By optimizing the loss function, the weights of our network f_θ will be updated accordingly.

Inference Process Following DDIM (Song et al., 2021), the reverse diffusion process is defined as a non-Markovian chain to achieve inference acceleration. An arithmetic sequence τ of length σ is predefined as $[1, ..., T]$ and D purely noised blocks $x_T \in \mathbb{R}^{D\times 5}$ are sampled from the Gaussian distribution. Modified from DDIM, we have progressive denoising as follows:

$$z_{\tau_{i-1}} = \sqrt{\bar{\alpha}_{\tau_{i-1}}}\hat{z}_0 + \sqrt{1-\bar{\alpha}_{\tau_{i-1}}}\frac{z_{\tau_i} - \sqrt{\bar{\alpha}_{\tau_i}}\hat{z}_0}{\sqrt{1-\bar{\alpha}_{\tau_i}}} \tag{3}$$

where $\hat{z}_0$ is predicted by f_θ, with the index i traversing from σ to 1. After σ iterations, $z_0 \in \mathbb{R}^{D\times 5}$ is recovered from the noise distribution. Note that D is a hyperparameter supposedly larger than the ground truth block number, and thus the filtration of predicted D blocks aims to minimize their divergence from the ground truth. Hence, blocks with the sum predicted probability below the threshold φ are discarded. [2]

3.3 Model Structure

As shown in Figure 2, our model architecture consists of three parts: Representation Encoder, Edge Predictor, and Level Predictor. Accepting one sentence, noised blocks (with timestep t) as inputs, the model network f_θ generates the predicted blocks $\hat{z}_0$ appropriately.

3.3.1 Representation Encoder

Given an input sentence $\mathcal{S} = \{x_1, x_2, ..., x_L\}$ composed of L words or indexes, here our sentence encoder consists of a pre-trained BERT (Devlin et al., 2019) and a bi-directional LSTM (Lample et al., 2016). Utilizing our encoder, token embeddings along with positional embeddings as the input are transformed into contextualized sentence representation $\mathbf{R}_H \in \mathbb{R}^{L\times d}$. Then the inner span tokens are extracted from the word indexes indicated by

[2] The probabilities, including $\mathbf{P}^\eta$ and $\mathbf{P}^\nu$, will be explained in Section 3.3.2.

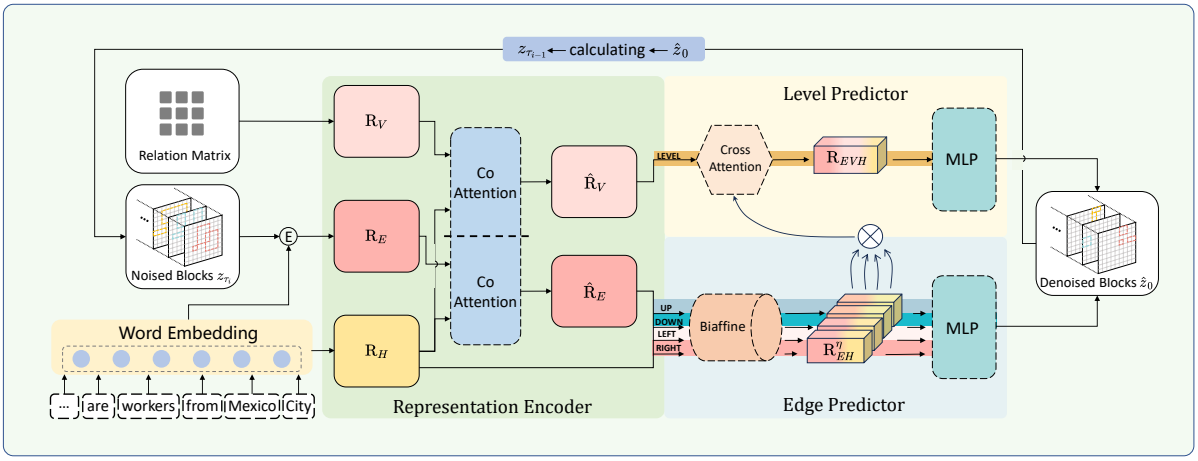

Figure 2: The overview model structure of IPED. To enhance the illustration of the workflow, we utilize three different colors to denote various feature representations: Pink for level information, Yellow for sentence information, and Red for edge information. Ⓔ represents the encoding of $\mathbf{R}_E$. $\otimes$ denotes the maxpooling operation. To simplify the illustration, the four Biaffine modules are integrated into one in this overview. To better display the reverse process as in Figure 1, a reverse-flow arrow is used to symbolize progressive denoising.

the edges of our blocks, yielding the edge representation $\mathbf{R}_E \in \mathbb{R}^{N\times d}$ following mean-pooling. Differently, the level representation $\mathbf{R}_V \in \mathbb{R}^{N\times d}$ is derived directly from an embedding relation matrix $\mathrm{R} \in \mathbb{R}^{K\times d}$, where each row represents a distinct relation type and K denotes the total number of predefined relation types. This matrix is regarded as a trainable parameter set in our model.

To better fuse both edge representation and level representation with contextualized information, we utilize the hierarchical Co-Attention mechanism in our model, which is proven to be effective with multimodal data (Chen et al., 2021). Among the two Parallel Co-Attention modules in our model, we illustrate one of them as an example, which attends to the sentence representation $\mathbf{R}_H$ and the edge representation $\mathbf{R}_E$ simultaneously. An affinity matrix $\mathbf{C} \in \mathbb{R}^{L\times N}$ that transforms sentence attention space into edge attention space, and the attention score vector $\mathbf{a}^e \in \mathbb{R}^N$ that optimizes the affinity, are calculated as follows:

$$\mathbf{C} = \tanh\left(\mathbf{R}_H^T \mathbf{W}_b \mathbf{R}_E\right) \tag{4}$$

$$\mathbf{H}^e = \tanh\left(\mathbf{W}_e \mathbf{R}_E + \left(\mathbf{W}_h \mathbf{R}_E\right)\mathbf{C}\right) \tag{5}$$

$$\mathbf{a}^e = \mathrm{softmax}\left(\mathbf{w}_{he}^T \mathbf{H}^e\right) \tag{6}$$

where $\mathbf{W}_b \in \mathbb{R}^{d\times d}$, $\mathbf{W}_e \in \mathbb{R}^{k\times d}$, $\mathbf{W}_h \in \mathbb{R}^{k\times d}$, $\mathbf{w}_{he} \in \mathbb{R}^k$ are learnable parameters, $\mathbf{H}^e$ is the middle state. Finally, the edge attention vector $\hat{\mathbf{R}}_E \in \mathbb{R}^{N\times d}$ is calculated as the weighted sum of the edge features plus an additional sinusoidal embedding (Vaswani et al., 2017):

$$\hat{\mathbf{R}}_E = \mathbf{a}^e \mathbf{R}_E + \mathbf{E}_t \tag{7}$$

where $\mathbf{E}_t$ is the embedding of timestep t. Equally, the same operation is implemented to obtain the fused level representation $\hat{\mathbf{R}}_V \in \mathbb{R}^{N\times d}$.

3.3.2 Edge Predictor and Level Predictor

For the Edge Predictor, we employ Biaffine to acquire fine-grained fused representations, which is proposed for dependency parsing (Dozat and Manning, 2016) at the outset. Here we have four Biaffine for $\mathbf{R}^{\eta}_{EH}$ representations where $\eta \in \{u, d, l, r\}$ symbolizes four edges, respectively. $\mathbf{R}^{\eta}_{EH}$ is obtained as follows:

$$\begin{aligned}\mathbf{R}^{\eta}_{EH} &= \mathrm{Biaff}^{\eta}\left(\mathbf{R}_H, \hat{\mathbf{R}}_E\right) \\ &= \mathbf{R}_H^T \mathbf{U}_1^{\eta} \hat{\mathbf{R}}_E + \mathbf{U}_2^{\eta}\left(\mathbf{R}_H \oplus \hat{\mathbf{R}}_E\right) + \mathbf{b}^{\eta}\end{aligned} \tag{8}$$

where $\mathbf{U}_1^{\eta}$ and $\mathbf{U}_2^{\eta}$ donate two parameter matrices, $\mathbf{b}^{\eta}$ is the bias vector, $\oplus$ means concatenation. Then $\mathbf{R}^{\eta}_{EH}$ are put through four simple multiple-layer perceptrons with softmax layers to get the probabilities $\mathbf{P}^{\eta} \in \mathbb{R}^{N\times L}$ for four edges in blocks.

For the Level Predictor, a cross-attention layer is utilized to obtain the deep latent representation $\mathbf{R}_{EVH}$, incorporating edge-sentence embedding $\mathbf{R}^{\eta}_{EH}$ to level representation $\hat{\mathbf{R}}_V$. Specifically, $\mathbf{R}^{\eta}_{EH}$ undergoes a max-pooling operation to

Method	NYT*			WebNLG*			NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
GraphRel (Fu et al., 2019)	63.9	60.0	61.9	44.7	44.1	42.9	-	-	-	-	-	-
RSAN (Yuan et al., 2020)	-	-	-	-	-	-	85.7	83.6	84.6	80.5	83.8	82.1
TPLinker (Wang et al., 2020)	91.3	92.5	91.9	91.8	92.0	91.9	91.4	92.6	92.0	88.9	84.5	86.7
GRTE (Ren et al., 2021)	92.9	93.1	93.0	93.7	94.2	93.9	93.4	93.5	93.4	92.3	87.9	90.0
PRGC (Zheng et al., 2021)	93.3	91.9	92.6	94.0	92.1	93.0	93.5	91.9	92.7	89.9	87.2	88.5
EmRel (Xu et al., 2022)	91.7	92.5	92.1	92.7	93.0	92.9	92.6	92.7	92.6	90.2	87.4	88.7
RelU-Net (Zhang et al., 2022)	93.3	92.9	93.1	94.9	93.7	94.3	-	-	-	-	-	-
BiRTE (Ren et al., 2022)	92.2	93.8	93.0	93.2	94.0	93.6	91.9	**93.7**	92.8	89.0	89.5	89.3
OneRel (Shang et al., 2022)	92.8	92.9	92.8	94.1	94.4	94.3	93.2	92.6	92.9	91.8	90.3	91.0
RFBFN (Li et al., 2022b)	93.4	93.2	93.3	93.9	94.1	94.0	93.7	93.6	93.6	91.5	89.4	90.4
ODRTE (Ning et al., 2023)	93.5	**93.9**	93.7	94.6	95.1	94.9	94.2	93.6	93.9	92.8	92.1	92.5
IPED	**94.2**	93.5	**93.9**	**95.3**	**95.7**	**95.5**	**94.7**	93.4	**94.1**	**93.0**	**93.6**	**93.3**

Table 1: Main results of IPED and other baselines.

serve as the key and value tensors, while $\hat{\mathbf{R}}_V$ acts as the query tensor. Then the level probability $\mathbf{P}^v \in \mathbb{R}^{N \times K}$ is determined using a multilayer perceptron, followed by a softmax layer.

3.3.3 Loss Function

In conjunction with the predicted probabilities above, the Log-Likelihood Function is maximized to train our model parameters. As N blocks are generated during training, yet only M ground truth blocks exist, we solve the optimal match via the Hopcroft-Krap algorithm (Carraresi and Sodini, 1986). Our objective function can be defined as follows:

$$\begin{aligned}\mathcal{L} = -\sum_{i=1}^{N}\Big[&\beta_1 \sum_{\eta \in \{u,d\}} \log \mathbf{P}_i^{\eta}\big(\xi^{\eta}(i)\big) \\ &+ \beta_2 \sum_{\eta \in \{l,r\}} \log \mathbf{P}_i^{\eta}\big(\xi^{\eta}(i)\big) \\ &+ \beta_3 \log \mathbf{P}_i^{v}\big(\xi^{v}(i)\big)\Big]\end{aligned} \tag{9}$$

where $\xi(i)$ represents the ground truth edges and level of the i-th block, $\beta_1, \beta_2, \beta_3$ are the hyperparameters for the weights of each prediction part.

4 Experiments

4.1 Datasets

Following previous works (Shang et al., 2022; Ning et al., 2023), we evaluate our model on two well-known datasets NYT (Riedel et al., 2010) and WebNLG (Gardent et al., 2017). The NYT dataset is extracted using the distantly supervised method from New York Times news articles, while the WebNLG dataset was originally designed for Natural Language Generation. Each dataset exists in two versions: one is annotated with the whole entity span, and the other is annotated with the last word of entities. For clarity, we mark the fully annotated version as NYT and WebNLG, and the simpler annotated version as NYT* and WebNLG*, respectively. Following prior works, we split the test set of each dataset based on the number of triples and the overlapping pattern in each sentence.

4.2 Evaluation Metrics

For a fair comparison with prior works mentioned above, we report standard micro Precision (Prec.), Recall (Rec.), and F1-score (F1.) as our three evaluation metrics. Meanwhile, we implement distinct matching rules for each version of the datasets. In the case of NYT and WebNLG datasets, an extracted relational triple is regarded correct only if all words of both entities and the relation type precisely align with the ground truth. For NYT* and WebNLG* datasets, only the last words of two entities and the relation are required to be correct.

4.3 Implementation Details

To make a fair comparison, we utilize the cased base version of BERT (Devlin et al., 2019) as our pretrained model. The AdamW optimizer (Loshchilov and Hutter, 2019) is employed with a learning rate of $3e$-5. The hidden size of our cross-attention and biaffine modules is configured

Model	NYT*								WebNLG*							
	Normal	SEO	EPO	Q=1	Q=2	Q=3	Q=4	Q≥5	Normal	SEO	EPO	Q=1	Q=2	Q=3	Q=4	Q≥5
GRTE	91.1	94.4	95.0	90.8	93.7	94.4	96.2	93.4	90.6	94.5	96.0	90.6	92.5	96.5	95.5	94.4
PRGC	91.0	94.0	94.5	91.1	93.0	93.5	95.5	93.0	90.4	93.6	95.9	89.9	91.6	95.0	94.8	92.8
RFBFN	91.2	95.2	95.6	91.4	**93.8**	94.8	96.4	93.9	91.0	94.6	96.5	90.8	92.6	96.6	94.7	94.5
ODRTE	**91.3**	95.7	95.9	91.3	93.4	94.6	96.9	95.3	92.1	95.4	95.9	91.1	93.5	95.9	96.1	95.1
IPED	91.0	**95.7**	**96.0**	**91.5**	93.2	**94.9**	**97.3**	**95.4**	**92.1**	**95.6**	**96.9**	**91.8**	**94.2**	**96.8**	**96.7**	**96.0**

Table 2: F1 score on sentences with different overlapping patterns and different triple numbers. Q stands for the number of triples in a sentence.

to 1024. A warm-up learning rate scheduler, with a 0.1 ratio and a maximum gradient normalization of 1.5, is configured for the training process. Regarding the diffusion setting, the total timestep T is set to 1000, the sampling timestep σ to 10, and the number of denoising blocks D to 30. The sum threshold φ for the edges and level probabilities is established at 4.

4.4 Main Results

Table 1 presents the performance comparison between our IPED and various baselines across four benchmarks. It can be seen that our model, IPED, outperforms all the baselines and achieves state-of-the-art performance, even when compared to the strongest explicit table-filling baseline ODRTE (Ning et al., 2023) and the leading multi-task joint framework RFBFN (Li et al., 2022b). This proves the dramatic efficacy of our implicit perspective and denoising diffusion strategy.

Compared with the best baseline ODRTE, our IPED achieves a 0.2 absolute improvement in F1-score on both NYT and NYT*. It is worth noticing that, a significant improvement, 0.8 and 0.6 gains in F1-score, is achieved on WebNLG and WebNLG* respectively, whereas many models (Wang et al., 2020; Gao et al., 2023) blame their poor performance on the complexity arising from hundreds of predefined relation types. We attribute our advancement on large-relation datasets to block-level progressive refinement; specifically, our block-denoising diffusion model allows fine-tuned block denoising across various levels of the tables.

The results on NYT and WebNLG reveal that our IPED outperforms OneRel (Shang et al., 2022) by 1.2% and 2.3%, and GRTE (Ren et al., 2021) by 0.7% and 3.3% in terms of F1-score, respectively. This demonstrates that the implicit table-filling scheme can immensely avoid interruptions caused by redundant negative tagging, which otherwise leads to negative bias. This improvement highlights two key advantages of our approach: the capability to recognize all potential triples and the proficiency in avoiding error association during decoding.

4.5 Performance on Complex Scenarios

To validate the ability of our model to handle diverse overlapping patterns and multiple triples, we conduct further experiments on NYT* and WebNLG*. As indicated in Table 2, our proposed IPED model surpasses nearly all baselines on both datasets, with the exception of two scenarios on NYT* when Q equals 2 and when there is no overlap. In complex scenarios, such as multiple triples within a single sentence, the performance of IPED turns out to be exceptional, surpassing four state-of-the-art models. The reason behind this is that our decoding scheme, the Parallel Boundary Emitting Strategy (PBES), has the capacity to accurately map our blocks into ground truth triples. This contrasts with previous decoding algorithms in explicit table-filling methods (Ren et al., 2021), which often incorrectly decode triples due to error association.

4.6 Computational Efficiency

To evaluate the computational efficiency of our IPED, we conduct further experiments with respect to *Training Time*, *GPU Memory*, *Inference Time*, and *F1-score* on NYT and WebNLG. As demonstrated in Table 3, we selected two robust baselines, GRTE and OD-RTE, for comparison. To verify the impact of the sampling timestep, we execute IPED with varying τ values. It can be seen that when $\sigma = 5$, the inference speed of IPED is more than double that of GRTE, and it requires the least GPU memory compared to both baselines. Due to the inherent nature of diffusion training, the training time of our model is not the shortest, falling be-

Model	NYT				WebNLG			
	Training Time	GPU Mem	Infer. Time (1/8)	F1	Training Time	GPU Mem	Infer. Time (1/8)	F1
GRTE	931†	18771†	44.1 / 9.6	93.4	118†	15345†	62.4 / 15.6	90.0
OD-RTE	**798**†	8372†	38.3 / 8.4	93.9	**70**†	7515†	51.0 / 12.8	92.5
IPED$_{[\sigma=5]}$	887	5636	**22.1 / 4.7**	94.0	102	3778	**30.1 / 7.7**	93.1
IPED$_{[\sigma=10]}$	887	5636	26.6 / 5.8	94.1	102	3778	35.5 / 8.7	93.3
IPED$_{[\sigma=15]}$	887	**5636**	33.4 / 7.2	**94.2**	102	**3778**	40.6 / 10.2	**93.4**

Table 3: Comparison of model efficiency. Training Time means the time (seconds) to train one epoch. GPU Mem stands for memory (MB) occupation during inference with the batch size of 8, and Infer. Time (1/8) donates the time (ms) to process each sentence with the batch sizes of 1 and 8, respectively. The superscript † indicates the results reported by OD-RTE. All experiments are conducted on a single GeForce RTX 3090 with default configuration.

tween OD-RTE and GRTE. Nevertheless, our IPED achieves a superior F1-score and greater inference efficiency. We conjecture the reasons might be our implicit table-filling strategy, which is exempt from redundant tagging, and the non-Markovian process employed during sampling.

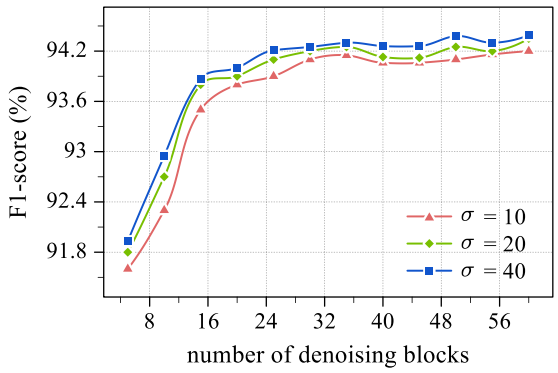

Figure 3: Performance of IPED with different number of denoising blocks D in terms of F1-score on NYT.

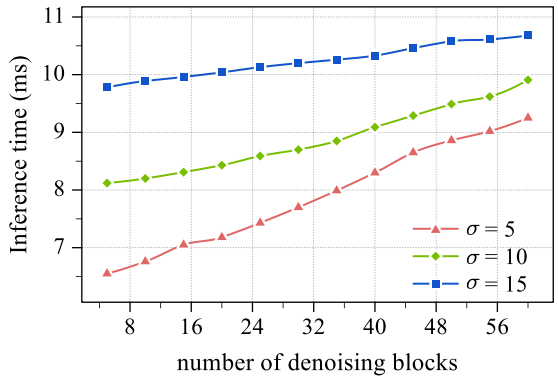

Figure 4: Performance of IPED with different number of denoising blocks D in terms of inference time on WebNLG. Note that the batch size is 8 during inference.

4.7 Analysis on Sampling Number

In the denoising inference process, the number of denoising blocks, denoted as D, is a crucial parameter. We conducted additional experiments on it with different sampling timestep σ to evaluate its impact on F1-score and inference time. As depicted in Figure 3, the F1-score decreases sharply when D is less than 15 and remains stable when D exceeds 25. It can be observed from Figure 4 that the inference time increases with larger D values, especially when σ is relatively small. Regarding the sampling timestep σ, these two figures indicate that a larger σ brings about a higher F1-score but also increases inference time. To balance the F1-score and inference time, we set D at 30 and σ at 10 as our standard configuration. Consequently, our IPED is capable of properly covering all potential blocks, thereby enhancing the recall rate while ensuring optimal inference time for practical applications.

5 Conclusion

This paper proposes an implicit approach to relational triple extraction, diverging from the explicit tagging methods of prior table-filling methods, thereby addressing several prevailing issues. Via denoising the edges and levels of noisy blocks, our introduced block-denoising diffusion model incrementally generates ground truth blocks, which can be swiftly and precisely converted into triples with our decoding algorithm PBES. Moreover, our network architecture incorporates beneficial modules such as Co-Attention and Biaffine, which promote the fusion of diverse representations. Experimental results on public datasets demonstrate that our IPED exceeds the performance of state-of-the-art (SoTA) models, while also achieving significantly faster inference speeds.

Limitations

Two limitations of IPED warrant discussion. Firstly, IPED exhibits a substantial increase in training time consumption compared to some models, as detailed in Section 4.6. This can be attributed to the extensive denoising timestep required for training, leading to slow and fluctuating convergence, thereby necessitating a greater number of training epochs. Secondly, the application of our implicit perspective is currently limited to relational triple extraction. Such perception holds potential for broader application in information extraction tasks such as document-level relation extraction and event extraction, addressing the issue of redundant negative tagging inherent in table-filling. These possibilities will be explored in our future work.

References

Guanqun Bi, Lei Shen, Yanan Cao, Meng Chen, Yuqiang Xie, Zheng Lin, and Xiaodong He. 2023. DiffusEmp: A diffusion model-based framework with multi-grained control for empathetic response generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2812–2831, Toronto, Canada. Association for Computational Linguistics.

Paolo Carraresi and Claudio Sodini. 1986. An efficient algorithm for the bipartite matching problem. *European Journal of Operational Research*, 23(1):86–93.

Richard J. Chen, Ming Y. Lu, Wei-Hung Weng, Tiffany Y. Chen, Drew FK. Williamson, Trevor Manz, Maha Shady, and Faisal Mahmood. 2021. Multimodal co-attention transformer for survival prediction in gigapixel whole slide images. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3995–4005.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. *ArXiv*, abs/1611.01734.

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.

Chen Gao, Xuan Zhang, LinYu Li, JinHong Li, Rui Zhu, KunPeng Du, and QiuYing Ma. 2023. Ergm: A multi-stage joint entity and relation extraction with global entity match. *Knowledge-Based Systems*, 271:110550.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2023. Diffuseq-v2: Bridging discrete and continuous text spaces for accelerated seq2seq diffusion models.

Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. 2023. DiffusionBERT: Improving generative masked language models with diffusion models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4521–4534, Toronto, Canada. Association for Computational Linguistics.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 917–928, Vancouver, Canada. Association for Computational Linguistics.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2021. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022a. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems*, volume 35, pages 4328–4343. Curran Associates, Inc.

Zhe Li, Luoyi Fu, Xinbing Wang, Haisong Zhang, and Chenghu Zhou. 2022b. RFBFN: A relation-first blank filling network for joint relational triple extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 10–20, Dublin, Ireland. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

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.

Jinzhong Ning, Zhihao Yang, Yuanyuan Sun, Zhizheng Wang, and Hongfei Lin. 2023. OD-RTE: A one-stage object detection framework for relational triple extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11120–11135, Toronto, Canada. Association for Computational Linguistics.

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.

Feiliang Ren, Longhui Zhang, Xiaofeng Zhao, Shujuan Yin, Shilei Liu, and Bochao Li. 2022. A simple but effective bidirectional framework for relational triple extraction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 824–832, New York, NY, USA. Association for Computing Machinery.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part III*, ECML PKDD'10, page 148–163, Berlin, Heidelberg. Springer-Verlag.

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.

Yongliang Shen, Xinyin Ma, Yechun Tang, and Weiming Lu. 2021. A trigger-sense memory flow framework for joint entity and relation extraction. In *Proceedings of the Web Conference 2021*, WWW '21, page 1704–1715, New York, NY, USA. Association for Computing Machinery.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. DiffusionNER: Boundary diffusion for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3875–3890, Toronto, Canada. Association for Computational Linguistics.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising diffusion implicit models. In *International Conference on Learning Representations*.

Changzhi Sun, Yeyun Gong, Yuanbin Wu, Ming Gong, Daxin Jiang, Man Lan, Shiliang Sun, and Nan Duan. 2019. Joint type inference on entities and relations via graph convolutional networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1361–1370, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Hao Zhou, Lei Li, and Junchi Yan. 2021. UniRE: A unified label space for entity relation 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 220–231, Online. Association for Computational Linguistics.

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, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020a. 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.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020b. 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.

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.

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 *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4054–4060. International Joint Conferences on Artificial Intelligence Organization. Main track.

Mohamad Zamini, Hassan Reza, and Minou Rabiei. 2022. A review of knowledge graph completion. *Information*, 13(8).

Yunqi Zhang, Yubo Chen, and Yongfeng Huang. 2022. RelU-net: Syntax-aware graph U-net for relational triple extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4217, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. 2023. Unleashing text-to-image diffusion models for visual perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5729–5739.

Hengyi Zheng, Rui Wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Xu Ming, and Yefeng Zheng. 2021. 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.

A Ablation Study

Ablation experiments are conducted to explore the contributions of the primary components within the network architecture and the effectiveness of level diffusion, as shown in Table 4. Observations reveal that removing any of the three components leads to a relative performance drop. Each of these three components is a critical part for representation construction, with the Co-Attention module having the most influence. Upon replacing the Co-Attention module with the simple addition of two input representations, a 1.2% F1 decline is observed. The experiments indicate that all three modules in our network play a crucial role in recovering blocks from noise.

It is noteworthy that the performance decreases by 2.2% when Level is omitted. This implies that IPED abandons the denoising diffusion process at the block Level, transitioning the task from three-dimensional to two-dimensional denoising. Specifically, noisy blocks are distributed across each level of the three-dimensional tables, with each block constrained to denoising at a specific level, thus precluding the possibility of progressive refinement with the block level. Thus it can be concluded that block-level denoising is crucial for the effectiveness of our block-denoising diffusion model in identifying triple relations, particularly in large-relation datasets like WebNLG.

B Dataset Statistics

The statistical details of the two datasets are displayed in Table 5.

Model	P	R	F
IPED	**93.0**	**93.6**	**93.3**
w/o Co-Attention	91.9	92.2	92.1
w/o Biaffine	92.2	93.0	92.6
w/o Cross Attention	92.1	92.5	92.3
w/o Level	90.6	91.6	91.1

Table 4: Ablation study on WebNLG dataset.

Dataset	Sentences			Details of test set								
	Train	Valid	Test	Normal	SEO	EPO	SOO	Q=1	Q=2	Q>2	Relations	Triples
NYT	56196	5000	5000	3071	1273	1168	117	3089	1047	864	24	8616
NYT*	56195	4999	5000	3266	1297	978	45	3244	1045	711	24	8110
WebNLG	5019	500	703	239	448	6	85	256	175	272	216	1607
WebNLG*	5019	500	703	245	457	26	84	266	171	266	171	1591

Table 5: Statistics of datasets used in our experiments. Q represents the number of triples in a sentence. Note that a single sentence can simultaneously contain SEO, EPO and SOO overlapping patterns.