Contrastive Triple Extraction with Generative Transformer

Hongbin Ye[1,2*], **Ningyu Zhang**[1,2*†], **Shumin Deng**[1,2], **Mosha Chen**[3], **Chuanqi Tan**[3],
Fei Huang[3], **Huajun Chen**[1,2†]
[1] Zhejiang University [2] AZFT Joint Lab for Knowledge Engine [3] Alibaba Group
{yehb,zhangningyu,231sm,huajunsir}@zju.edu.cn, {chenmosha.cms,chuanqi.tcq,f.huang}@alibaba-inc.com

Abstract

Triple extraction is an essential task in information extraction for natural language processing and knowledge graph construction. In this paper, we revisit the end-to-end triple extraction task for sequence generation. Since generative triple extraction may struggle to capture long-term dependencies and generate unfaithful triples, we introduce a novel model, contrastive triple extraction with a generative transformer. Specifically, we introduce a single shared transformer module for encoder-decoder-based generation. To generate faithful results, we propose a novel triplet contrastive training object. Moreover, we introduce two mechanisms to further improve model performance (i.e., batch-wise dynamic attention-masking and triple-wise calibration). Experimental results on three datasets (i.e., NYT, WebNLG, and MIE) show that our approach achieves better performance than that of baselines.

Introduction

Triple extraction is an essential information extraction task for natural language processing (NLP) and knowledge graph (KG), which is used to detect pairs of entities and their relations from unstructured text. Consider this sentence: "*Paris is known as the romantic capital of France*." From this, an ideal triple extraction would comprise ⟨*Paris, Capital_of, France*⟩, in which *Capital_of* is the relation of *Paris* and *France*.

Researchers have proposed pipeline approaches in the past (Lample et al. 2016; Zeng et al. 2015) in which they typically deconstructed the triple extraction problem into two separate tasks: named-entity recognition (NER) (used to extract entities) and relation classification. Thus, they first recognized the entities; then, they predicted their relationships. Unfortunately, this and similar pipeline approaches suffer drawbacks (Roth and Yih 2007) in that they omit the evident correlations between entity recognition and relation extraction tasks, resulting in error propagation.

Recently, several neural-network-based models (Zeng et al. 2018a) have been proposed to jointly extract entities and relations from sentences. These models use a parameter-sharing mechanism to extract entities and relations from the same network. Apart from those approaches, Zeng et al. (2018b) proposed a recurrent neural-network-based encoder-decoder model (i.e., CopyRE) to extract triples with overlapping entities. Such end-to-end generative triple extraction not only directly obtain the triples and mitigate the error propagation issue, but also enable the generation of out of domain entities and relations in a T5-style (Raffel et al. 2019) (text-to-text). Besides, Zeng, Zhang, and Liu (2020) proposed a multi-task learning framework equipped with a copy mechanism (i.e., CopyMTL) to allow the prediction of multi-token entities. Nayak and Ng (2019) introduced a representation scheme for triples and a pointer-network-based decoding approach, which further improved the performance of CopyRE.

Input	The *United States* President *Trump* was raised in the borough of *Queens* in *New York City*, and lived there until age 13.
Output	Trump→president→of→United→States→ [S2S_SEQ]→Trump→born→in→Queens→ [S2S_SEQ]→Trump→live→in→Queens
Gold	(Trump, president_of, United States) (Trump, born_in, Queens) (Trump, live_in, Queens)
Negative	(Trump, president_of, Queens) (Trump, born_in, 13) (Trump, live_in, 13)

Table 1: Contrastive triple extraction as sequence generation. We encourage the model to generate gold triples and does not generate negative ones.

Encoder-decoder models are powerful tools that have seen success in many NLP tasks, including machine translation (Cho et al. 2014), and open information extraction (Zhang, Duh, and Van Durme 2017). Although significant progress has been achieved, there remain two key problems with the existing methods. **First**, owing to the intrinsic shortfalls of recurrent neural networks (RNN), they cannot capture long-term dependencies, which results in the loss of important information otherwise reflected in the sentence. Such a drawback prevents the model from being applied to longer texts.

[*] Equal contribution and shared co-first authorship.

[†] Corresponding author.

Second, there is a scarcity of work that has focused on generating faithful triples. As a previous study (Zhu et al. 2020) indicated, a sequence-to-sequence architecture can generate unfaithful sequences that create contradictions of meaning. For example, given the sentence "The *United States* President *Trump* was raised in the borough of *Queens* in *New York City*, and lived there until age 13," the model could generate the fact "(Trump, born_in, Queens)." Although logically true, we cannot find direct evidence from the given sentence to support it.

To address these issues, we introduce a framework of **C**ontrastive triple extraction with **G**enerative **T**ransformer (CGT), which is a single shared transformer module with a triplet contrastive object that supports encoder-decoder generation. To begin with, we concatenate the input sequence with the target sequence using a separator token and leverage partial causal masking (Du, Rush, and Cardie 2020) to distinguish the encoder-decoder representations. Our model requires no additional parameters beyond those of the pretrained model. Then, we introduce a novel triplet contrastive learning object, which utilizes ground-truth triples as positive instances and leverages random token sampling to construct corrupt triples as negative instances. To jointly optimize the triple generation and contrastive object, we introduce a batch-wise dynamic attention-masking mechanism, which allows us to dynamically choose different objects and jointly optimize tasks. Lastly, we introduce a novel triplewise calibrating algorithm to filter out any remaining false triples in the inference stage.

The contributions of this work are as follows:

- We revisit triple extraction as a sequence generation task and introduce a novel CGT model. In light of the added extraction capability, CGT requires no additional parameters beyond those found in the pre-trained language model.
- We introduce two mechanisms to further improve model performance (i.e., batch-wise dynamic attention-masking and triple-wise calibration). The first enables joint optimization of different objects, and the second ensures faithful inference.
- We evaluate CGT on three benchmark datasets. Our model empirically outperforms other substantially strong baseline models. We also demonstrate that CGT is better than existing triple extraction approaches at capturing long-term dependencies, thus, achieving better performance with long sentences.

Related Work

Triple Extraction

Two main methods have been proposed for triple extraction: pipeline (Nadeau and Sekine 2007; Bunescu and Mooney 2005; Lin et al. 2016; Lin, Liu, and Sun 2017; Li et al. 2020; Wang et al. 2020) and joint learning (Miwa and Bansal 2016; Katiyar and Cardie 2017; Cao et al. 2017a; Zhang et al. 2020a; Dai et al. 2019a). A pipeline method first extracts entities, then it identifies their relations (Hendrickx et al. 2019; Zeng et al. 2015; Wu et al. 2021). Although pipeline models have achieved great progress (Zhang et al. 2018; He et al. 2018; Zhang et al. 2019a, 2020c), they introduce an error propagation problem (Li and Ji 2014), which does harm to the overall performance.

Because joint learning can implicitly model correlations between tasks, many approaches have been proposed. Bekoulis et al. (2018) formulated the triple extraction task as a multi-head selection problem. Takanobu et al. (2019) proposed a hierarchical reinforcement-learning framework for triple extraction. Chen et al. (2019) utilized triplet attention to exploit connections between the relation and its corresponding entity pairs. Dai et al. (2019b) introduced a position-attention mechanism to produce different tag sequences for triple extraction.Wei et al. (2020a) revisited the relational triple extraction task and proposed a novel cascade binary-tagging framework. Apart from those approaches, Zeng et al. (2018a) proposed CopyRE, a joint model based on a copy mechanism, which converted the joint extraction task into a triplet-generation task. Other researchers introduced multiple strategies, such as multi-task learning (Zeng, Zhang, and Liu 2020) and one-word generation (Nayak and Ng 2019) to improve CopyRE. For the first time, we utilize the transformer as an encoder-decoder architecture to extract triples from sentences.

Natural Language Generation

Natural language generation has been intensively studied in the recent literature. Most models employed an encoder-decoder architecture (i.e., seq2seq) using RNNs (Schuster and Paliwal 1997; Zhang et al. 2020b; Krause et al. 2020). Recently, owing to the powerful representation ability of transformers, several researchers have introduced transformer-based natural language generation methods. Gu, Wang, and Zhao (2019) developed the Levenshtein transformer, a new partially autoregressive model, which is devised for a more flexible and amenable sequence generation. Chen et al. (2020) present a novel approach, Conditional Masked Language Modeling (C-MLM), to enable the finetuning of BERT (Devlin et al. 2019) on target generation tasks. Dong et al. (2019) proposed a new unified pre-trained language model with different masking strategies, which can be used for both language understanding and generation. Du, Rush, and Cardie (2020) proposed a generative transformer-based encoder-decoder framework for document-level information extraction.

Since the generation procedure was unconditional, it was non-trivial to judge the faithfulness of the generated sequence. Zhang et al. (2019b) approached the factual correctness problem in the medical domain, where the space of facts was limited and could be depicted with a descriptor vector. Cao et al. (2017b) extracted relational information from an article and mapped it to a sequence as input to the encoder. The decoder then attended to both article tokens and their relations. Gunel et al. (2020) employed an entity-aware transformer structure to boost the factual correctness of abstractive summarization, where the entities came from the Wikidata knowledge graph. By comparison, our model utilizes contrastive learning to encourage the model to implicitly generate faithful triples.

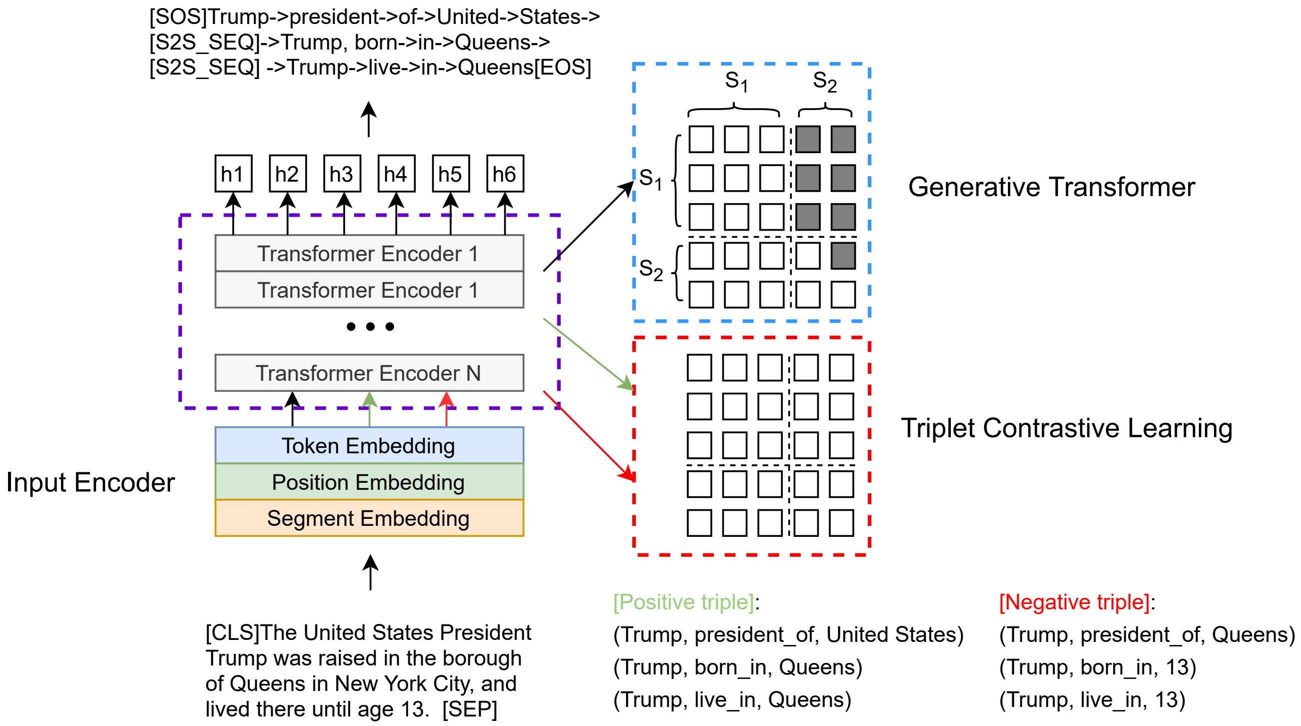

[SOS]Trump->president->of->United->States->
[S2S_SEQ]->Trump, born->in->Queens->
[S2S_SEQ] ->Trump->live->in->Queens[EOS]
h1
h2
h3
h4
h5
h6
Transformer Encoder 1
Transformer Encoder 1
Transformer Encoder N
Token Embedding
Position Embedding
Segment Embedding
Input Encoder
[CLS]The United States President Trump was raised in the borough of Queens in New York City, and lived there until age 13. [SEP]
S1
S2
S1
S2
Generative Transformer
Triplet Contrastive Learning
[Positive triple]:
(Trump, president_of, United States)
(Trump, born_in, Queens)
(Trump, live_in, Queens)
[Negative triple]:
(Trump, president_of, Queens)
(Trump, born_in, 13)
(Trump, live_in, 13)

Figure 1: The architecture of **C**ontrastive triple extraction with **G**enerative **T**ransformer (CGT). The top-right component refers to the generative transformer, and the bottom-right component represents triplet contrastive learning. Those two parts are optimized jointly. The left is the input encoder (best viewed in color).

Overview

Preliminary

We treat triple extraction as a sequence-to-sequence task to better model the cross dependencies between entities and relations. We define the input text and output triples as source and target sequence. As shown in Figure 1, the source sequence simply consists of the tokens of the input sentence like "[CLS] The United States President Trump was raised in the borough of Queens ...[SEP]". We concatenate the triples for each entity/relation separated by a special token token [S2S_SEQ] as the target sequence. We also add the beginning ([SOS]) and end ([EOS]) tokens for each target sequence as:

$$\begin{gathered}
[\text{SOS}]h^{(1)}, r^{(1)}, t^{(1)} \dots [\text{S2S\_SEQ}] \\
h^{(2)}, r^{(2)}, t^{(2)} \dots [\text{S2S\_SEQ}] \\
h^{(3)}, r^{(3)}, t^{(3)} \dots [\text{S2S\_SEQ}] \\
\dots \\
h^{(N)}, r^{(N)}, t^{(N)} \dots [\text{EOS}],
\end{gathered}$$

where h^i, r^i, and t^i refer to the i-th generated head entity, relation, and tail entity.

Framework

We denote the sequence of input source tokens as x_0, x_1, ..., x_m and the sequence of target tokens as y_0, y_1, ..., y_n. Note that the generated tokens contain all extracted triples. Our model CGT consists of three components, as follows:

Input Encoder. We utilize the input representation which is the same as BERT (Devlin et al. 2019) and tokenize texts by WordPiece (Yonghui et al. 2016). We compute the representation by summing the corresponding token embedding, position embedding, and segment embedding.

Generative Transformer. We use partial causal masking to distinguish the encoder-decoder representations. For inference, we leverage the beam search (Wiseman and Rush 2016) to generate multiple triples.

Triplet Contrastive Learning. We introduce a triplet contrastive object to enhance the faithfulness of generated triples. We introduce a batch-wise dynamic attention masking mechanism for joint optimization. We also provide a triple-wise calibrating algorithm for the faithful triple generation.

Our Model

Input Encoder

Given the input text x, we add a special start-of-sequence token [SOS] at the beginning of the target input. We use the representation of the whole input for the output vector. Furthermore, we append a special token, namely, end-of-sequence [EOS], to the end of each output sequence. The

[EOS] token is used as a special token to terminate the decoding process for the triple generation.

The input representation is the same as the one used for BERT (Devlin et al. 2019). We tokenize the text to subword units using WordPiece (Yonghui et al. 2016). For example, the word, "forecasted," is split into "forecast" and "##ed," where "##" refers to the pieces belong to one word. We compute each input token vector representation by summing the corresponding token embedding, position embedding, and segment embedding.

Generative Transformer

We utilize a transformer architecture as a backbone to encode contextual features which is consist of stacked self-attention layers. In this paper, we use a 12-layer transformer architecture as a single shared transformer module for encoder-decoder-based generation. Having the input vectors, $\{\mathbf{s}_i\}_{i=1}^{|L|}$, we firstly feed them into $\mathbf{H}^0 = \left[\mathbf{s}_1, \cdots, \mathbf{s}_{|L|}\right]$. Then, we use the transformer to encode the input:

$$\mathbf{H}^l = \mathrm{Transformer}_l\left(\mathbf{H}^{l-1}\right). \tag{1}$$

There are multiple self-attention heads in each transformer block which are used to aggregate the output vectors of the previous layer. We compute the output of a self-attention head, A_l, in the l-th transformer layer as follows:

$$\mathbf{Q}_l = \mathbf{H}^{l-1}\mathbf{W}_l^Q, \quad \mathbf{K}_l = \mathbf{H}^{l-1}\mathbf{W}_l^K. \tag{2}$$

$$\mathbf{M}_{ij} = \begin{cases} 0, & \text{allow to attend} \\ -\infty, & \text{prevent from attending} \end{cases} \tag{3}$$

$$\mathbf{A}_l = \mathrm{softmax}\left(\frac{\mathbf{Q}_l\mathbf{K}^\top}{\sqrt{d_k}} + \mathbf{M}\right)\left(\mathbf{H}^{l-1}\mathbf{V}_l\right), \tag{4}$$

where $\mathbf{Q}_l, \mathbf{K}_l, \mathbf{V}_l \in \mathbb{R}^{d_h \times d_k}$ are matrices which are the projection of the the previous layer's output. The mask matrix, $\mathbf{M} \in \mathbb{R}^{|L| \times |L|}$, is aimed to control the context that can be attended by the token. Specifically, we leverage different mask matrices, $\mathbf{M}$, when computing its contextualized representation. As illustrated by the examples in Figure 1, for triple generation, we leverage partial causal masking, in which the upper right part is set to $-\infty$ to block attention from the source segment to the target segment; the left part of $\mathbf{M}$ is set to all 0s which indicates that the tokens is able to attend to the first segment. We utilize cross-entropy $\mathbf{loss}_{generation}$ to optimize the triple generation procedure. We also utilize masking strategies in which the elements of the mask matrix are all 0s for triplet contrastive learning. Details are provided in the next section. Formally, the generative transformer obtain contextualized representations and optimize the following object:

$$\begin{aligned} &\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_m, \hat{\mathbf{y}}_0 \ldots, \hat{\mathbf{y}}_n \\ &= \mathrm{Transformer}\left(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_m, \mathbf{y}_0, \ldots, \mathbf{y}_n\right) \end{aligned} \tag{5}$$

$$\mathbf{loss}_{generation} = \sum\left(\sum_1^m \mathbf{x}_i log(\hat{\mathbf{x}}_i) + \sum_1^n \mathbf{y}_i log(\hat{\mathbf{y}}_i)\right) \tag{6}$$

Triplet Contrastive Learning

The previous generation-based approach usually neglects the fact that triple should be faithful and consistent with the input sentence. For example, given the instance "Obama was born in Honolulu," we should engorge the model to generate triples like "(Obama, was_born, Honolulu)" rather than "(Obama, live_in, Honolulu)," though the latter may be correct but cannot be induced from the given sentence. Motivated by this, we introduce a triplet contrastive learning to enhance the faithfulness of generated triples.

To be specific, we leverage the triple contrastive learning as binary classification with all 0s masking. We use gold triples as positive instances and generate corrupt triples by replacing one entity with random tokens in the instances. We use those corrupt triples as negative instances. We concatenate the input sentence with only one triple as x_0, x_1, ..., $x_m[SEP], h^i, r^i, t^i$ and feed it into the input encoder. We utilize the representation of [CLS] with an MLP layer to compute classification logits z. We utilize cross-entropy for optimization with $\mathbf{loss}_{contrastive}$:

$$\mathbf{loss}_{contrastive} = \sum(\mathbf{z}_i^+ log(\hat{\mathbf{z}}_i^+) + (1 - \mathbf{z}_i^-) log((1 - \hat{\mathbf{z}}_i^-))) \tag{7}$$

where $\hat{\mathbf{z}}_i^+$ and $\hat{\mathbf{z}}_i^-$ are the positive and negative logits, respectively. Formally, the triplet contrastive learning algorithm for triple extraction is as follows:

Algorithm 1 Triplet Contrastive Learning

```
Require: Train instances $X = x_1, ..., x_N$, labels $Y = y_1, ..., y_N$, batch size $k$, $POS = \Phi$, $NEG = \Phi$, temperature $t$
while $i \leq N/k$ do
    batch = $[(x, y)_1, .., (x, y)_k]$
    for $(x, y)_j$ in batch do
        POS = decompose_triple($y_j$)
        for $pos$ in POS do
            neg = random_permute_entity(pos)
            l_pos = Contrastive_Classify(x,pos)
            l_neg = Contrastive_Classify(x,neg)
            z = cat([l_pos, l_neg], dim=1)
            labels = zeros(2)
            loss = CrossEntropyLoss(z/t, labels)
            loss.backward()
            update(Contrastive_Classifier.params)
return DataLoader
```

Training and Inference Details

During the training stage, the entities and relations are all tokens from the vocabulary, whereas [S2S_SEQ], [SOS], and [EOS] are all unused tokens (e.g., [unused1]). We split the entity and relation label mentions with different tokens during the data preprocessing procedure, meaning that the entity and relation may contain multiple tokens.

Note that triplet contrastive learning and triple generation are two different tasks, and optimizing them jointly is nontrivial, owing to the leakage of generated labels. For example, if we optimize generation and contrastive learning with the same instance, the model can see all of the tokens because of the all 0s masking. To address this issue, we introduce batch-wise dynamic attention masking. With this, we sample instances from a Bernoulli distribution as generation instances, and the rest is sampled as contrastive learning sentences. Formally, the algorithm is as follows:

Algorithm 2 Batch-wise Dynamic Attention Masking

```
Require: Train instances $X = x_1, ..., x_N$, labels $Y = y_1, ..., y_N$, negative instances $Y'$, batch size $k$ sampling ratio $\gamma$
while $i \leq N/k$ do
    old_batch = $[(x, y, y')_1, .., (x, y, y')_k]$
    for $(x, y, y')_j$ in old_batch do
        condition = Bernoulli($\gamma$)
        if condition == 1 then
            instance = Partial_Causal_Mask($(x, y, y')_j$)
        else
            instance = All_Zero_Mask($(x, y, y')_j$)
        batch ← batch ∩ instance
    DataLoader ← DataLoader ∩ batch
    batch = $\Phi$
return DataLoader
```

The overall optimization object is as follows:

$$\textbf{loss} = \textbf{loss}_{generative} + \alpha\textbf{loss}_{contrastive} \quad (8)$$

where α is the hyperparameter to balance different objects.

During the inference stage, we first generate triplet sequences via beam search (Wiseman and Rush 2016). Then, we introduce a triple-wise calibrating algorithm to filter-out unfaithful triples. We calculate the matching score with the contrastive classifier and filter out those triples with the $match\_score < \theta$. Besides, we also leverage heuristic rules to generate reasonable triples such as the relation should be followed by the head entities.

Experiment

Dataset

We conducted experiments on three benchmark datasets: New York Times (NYT), WebNLG[1], and MIE[2]. The NYT dataset is produced using a distant supervision method and is widely used for triplet extraction (Riedel, Yao, and McCallum 2010). It contains 56,195 sentences for training, 5,000 sentences for validation, and 5,000 sentences for test. The WebNLG dataset (Gardent et al. 2017) was used for natural language generation, but was later used for triplet extraction (Zeng et al. 2018a). It consists of 5,019/500/703 instances for training, validation, and testing, respectively. MIE (Zhang et al. 2020d) is a large-scale Chinese dialogue information extraction dataset for the medical domain. It contains 800 instances for training, 160 instances for validation, and 160 instances for testing. We used the original dataset splitting for NYT, WebNLG, and MIE. Detailed statistics of the three datasets are shown in Table 2.

Dataset	NYT	WebNLG	MIE
Domain	News	Web	Medical
Relation	24	246	343
Triplets	104,518	12,863	18,212

Table 2: Statistics of four datasets in the domain, the number of relation types, and the triple number.

Settings

We utilized *UniLM-base-uncased* for both English[3] and Chinese[4] datasets. We utilized Pytorch to implement our CGT model and conducted experiments using four Nvidia 1080-Ti graphical processing units. We employed Adam (Kingma and Ba 2014) as the optimizer. The initial learning rate was set to 2e-5, and we reduced the rate by 20% at every eight epochs. The batch size was 64 for English and 32 for Chinese, and the total number of epochs was 50 for all datasets. The beam size was set to 4, α was set to 0.1, γ was set to 0.2, and θ was set to 0.6. We carefully tuned the hypermeters on the valid set (Detailed search space in supplementary materials).

Baselines and Evaluation Metrics

We compared the performance of CGT with various baseline models and evaluated the performance with precision, recall, and F1 score. CGT(Random) and CGT(UniLM) refer to the model initialized randomly, and the model initialized with UniLM, respectively.

Generative Baseline Models:

CopyRE (Zeng et al. 2018a) is a Seq2Seq learning framework having a copy mechanism wherein multiple decoders are applied to generate triples to handle overlapping relations.

PNDec (Nayak and Ng 2019) provides two novel approaches using encoder-decoder architecture for triples having multiple tokens.

CopyMTL (Zeng, Zhang, and Liu 2020) proposes a multitask learning framework used to complete the entities.

Extractive Baselines:

Tagging (Zheng et al. 2017b) is an end-to-end method that uses a novel tagging scheme.

HRL (Takanobu et al. 2019) addresses relation extractions by regarding related entities as the arguments of the relation via hierarchical reinforcement learning.

MrMep (Chen et al. 2019) is an approach that utilizes triplet attention to exploit connections between relations and their corresponding entity pairs.

CasRel (Wei et al. 2020a) is an approach that models relations as functions, which map subjects to objects in a sentence.

Bi-LSTM (Zhang et al. 2020d) is a baseline approach that utilizes a bi-directional long-short term memory network for information extraction.

[1] https://github.com/weizhepei/CasRel

[2] https://github.com/nlpir2020/MIE-ACL-2020

[3] https://github.com/microsoft/unilm

[4] https://github.com/YunwenTechnology/Unilm

Model		NYT			WebNLG		
		P	R	F	P	R	F
Extractive	Tagging (Zheng et al. 2017a)	61.5	41.4	49.5	-	-	-
	HRL(Takanobu et al. 2019)	71.4	58.6	64.4	53.8	53.8	53.8
	MrMep (Chen et al. 2019)	77.9	76.6	77.1	69.4	77.0	73.0
	CasRel (Wei et al. 2020b)	89.7	89.5	89.6	93.4	90.1	91.8
Generative	CopyRE (Zeng et al. 2018a)	61.0	56.6	58.7	37.7	36.4	37.1
	PNDec (Nayak and Ng 2019)	80.6	77.3	78.9	38.1	36.9	37.5
	CopyMTL (Zeng, Zhang, and Liu 2020)	75.7	68.7	72.0	58.0	54.9	56.4
Ours	CGT(Random)	90.8	77.7	83.7	87.6	70.5	78.1
	CGT(UniLM)	**94.7***	84.2	89.1	**92.9***	75.6	83.4
	w/o contrastive	87.3	81.5	84.3	94.6	70.5	80.8

Table 3: Main results of NYT and WebNLG. The top section refers to the extractive models, the middle section indicates the generative approaches, the bottom is our model with different settings. * indicates $p_{value} < 0.01$ for a paired t-test evaluation.

Model	P	R	F1
Bi-LSTM	53.13	49.46	50.69
MIE-multi	70.24	64.96	66.40
CGT(random)	70.75	66.96	68.80
CGT(UniLM)	**80.53**	**78.83**	**79.42**

Table 4: Main results on the MIE dataset.

MIE-multi (Zhang et al. 2020d) is another baseline model that uses a max-pooling operation to obtain the final score, considering the turn-interaction.

Main Results

From Table 3, we observe that our approach achieved significant improvements compared with all generation-based baseline models for both NYT and WebNLG datasets. Our CGT model had a relative **10.2** F1 score improvement on NYT compared with PNDec, and a relative **27.0** F1 score improvement on NYT compared with CopyMTL, illustrating the power of our proposed model. Our approach also obtained comparable results compared with extractive models, such as CasRel. Note that the search space of the generative model was much larger than the extractive ones, which indicates that the generative model was challenging to optimize than extractive approaches. In contrast, generative methods can generate triples beyond the entity and relation domain, which is promising for the open domain setting. The empirical results reveal that the generative approach could obtain comparable performance with extractive models, motivating future research directions.

From Table 4, we observe that our approach achieved significant improvements (relative **13.02** F1 score) compared with all baselines on the MIE dataset. MIE is a dialogue-based information-extraction dataset that is challenging to optimize. Thus, we argue that our CGT can implicitly model the relations among entities, boosting performance.

Ablation Study

We conducted ablation studies further to demonstrate the efficacy of different strategies in our model. From Table 3, we notice that the performance decayed without contrastive object, which illustrates that triplet contrastive learning can enhance the faithfulness of generated triples, thus boosting the performance. We also observe that our approach with random initialization CGT(Random) achieves significantly better performance than generative baselines on all three benchmark datasets, which further indicates that our improvements are not only from the pre-trained language model but also the model architecture itself.

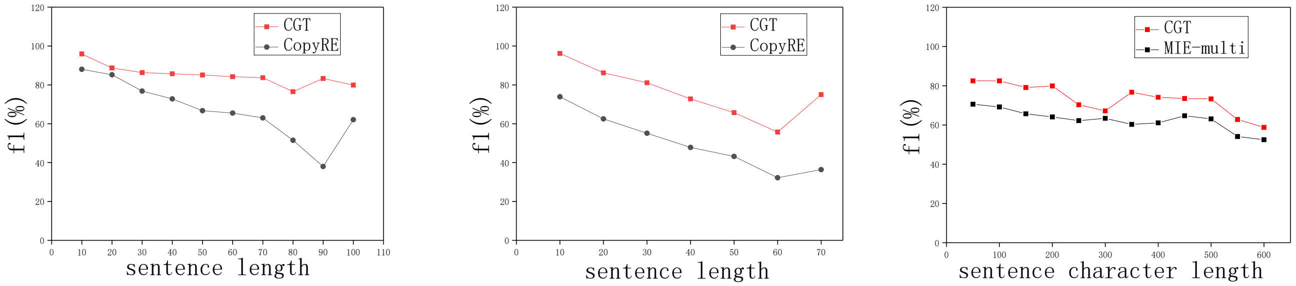
Analysis

To better analyze the performance of our proposed CGT model, we conducted a detailed analysis and attempted to answer the questions of whether CGT can capture long-term dependence or not. Intrusively, transformers having self-attention can better capture long-term dependencies than RNNs. To investigate this issue, we evaluated the instances at different lengths. From Figure 2, we notice that all models have a performance decay when the sentence length increases, which indicates that the sequence generation is challenging when the input sentence is long. We observe that our approach could obtain better performance than that of CopyRE when the sentence length increased. When the sentence was longer than 60, CopyRE archived worse performance, while CGT performed relatively better. This demonstrates that the proposed approach can capture long-term dependencies, compared with RNN-based approaches.

Error Analysis

To further analyze the drawbacks of our approach and promote future works of triple extraction, we select instances and conduct error analysis. We random select incorrect instances and classify them into three categories bellows, as shown in Table 5:

Distract Context. As instance #1 shows, we observe that our approach may fail to those ambiguous contexts that may



(a) NYT

(b) WebNLG

(c) MIE

Figure 2: Model performance #*sentence length*.

Instance
instance #1 Batchoy is originates from the Philippines and served as a soup.Its main ingredients are noodles, pork organs, vegetables, chicken, shrimp and beef. generated triple: ⟨*Batchoy, location, Philippines*⟩ ground truth: ⟨*Batchoy, country, Philippines*⟩
instance #2 Alan Shepard was a crew member of NASA operated Apollo 14 who died in California which is represented by Dianne Feinstein. generated triple: ⟨*Shepard, deathPlace, California*⟩ ground truth: ⟨*Allan Shepard, deathPlace, California*⟩
instance #3 Saranac Lake, which is served by Adirondack Regional Airport, is part of Harrietstown, Essex County, New York, US. generated triple: ⟨*Airport, cityServed, New York*⟩ ground truth: ⟨*Airport, cityServed, York*⟩

Table 5: Error anslysis.

be expressed in a similar context but differ only in the fine-grained type of entities. We argue that this may be caused by the unbalanced learning problems that models tend to judge the sentence with similar context to high-frequency relations.

Wrong Boundaries. As instance #2 shows, generated triples had incorrect boundaries, which indicates the difficulty of entity recognition during triple extraction. We argue that since our approach is an end-to-end generation method, it is challenging to capture fine-grained entity boundaries without sequence token information.

Wrong Triples. As instance #3 shows, many generated triples had entities that did not exist in the gold-standard set. Generally, this occurs with sentences having multiple triples. The WebNLG datasets are noisy, and several of its cases produced incorrect results. We leave this for future works with more suitable benchmarks.

Conclusion and Future Work

In this paper, we revisited triple extraction as a sequence generation task, which jointly extracts entities and relations. To address the long-term dependence and faithfulness issues, we proposed a novel CGT model to generate faithful triples. To the best of our knowledge, we are the first to integrate sequence generation with contrastive learning for information extraction, which may inspire future research directions and motivate new ideas. Experimental results on three datasets demonstrated the efficacy of our approach. In the future, we will utilize stronger transformer architectures, such as Longformer (Beltagy, Peters, and Cohan 2020) to generate relational knowledge from documents. We will also delve into injection ontology knowledge using condition generation methods. It will also be useful to apply our approach to other scenarios, such as event extractions.

Acknowledgments

We want to express gratitude to the anonymous reviewers for their hard work and kind comments. We thank Ning Ding for helpful discussions and feedback on this paper. This work is funded by 2018YFB1402800/NSFC91846204/NSFCU19B2027.

References

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

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* .

Bunescu, R. C.; and Mooney, R. J. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, 724–731. Association for Computational Linguistics.

Cao, Y.; Huang, L.; Ji, H.; Chen, X.; and Li, J. 2017a. Bridge Text and Knowledge by Learning Multi-Prototype Entity Mention Embedding. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1623–1633. Vancouver, Canada: Association for Computational Linguistics. doi:10.18653/v1/P17-1149. URL https://www.aclweb.org/anthology/P17-1149.

Cao, Z.; Wei, F.; Li, W.; and Li, S. 2017b. Faithful to the original: Fact aware neural abstractive summarization. *arXiv preprint arXiv:1711.04434* .

Chen, J.; Yuan, C.; Wang, X.; and Bai, Z. 2019. MrMep: Joint Extraction of Multiple Relations and Multiple Entity Pairs Based on Triplet Attention. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, 593–602.

Chen, Y.-C.; Gan, Z.; Cheng, Y.; Liu, J.; and Liu, J. 2020. Distilling Knowledge Learned in BERT for Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7893–7905.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* .

Dai, D.; Xiao, X.; Lyu, Y.; Dou, S.; She, Q.; and Wang, H. 2019a. Joint extraction of entities and overlapping relations using position-attentive sequence labeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6300–6308.

Dai, D.; Xiao, X.; Lyu, Y.; Dou, S.; and Wang, H. 2019b. Joint Extraction of Entities and Overlapping Relations Using Position-Attentive Sequence Labeling. *Proceedings of the AAAI Conference on Artificial Intelligence* 33: 6300–6308.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics. doi:10.18653/v1/N19-1423.

Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; and Hon, H.-W. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, 13063–13075.

Du, X.; Rush, A.; and Cardie, C. 2020. Document-level Event-based Extraction Using Generative Template-filling Transformers. *arXiv preprint arXiv:2008.09249* .

Gardent, C.; Shimorina, A.; Narayan, S.; and Perez-Beltrachini, L. 2017. Creating training corpora for nlg micro-planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*.

Gu, J.; Wang, C.; and Zhao, J. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems*, 11181–11191.

Gunel, B.; Zhu, C.; Zeng, M.; and Huang, X. 2020. Mind The Facts: Knowledge-Boosted Coherent Abstractive Text Summarization. *arXiv preprint arXiv:2006.15435* .

He, Z.; Chen, W.; Li, Z.; Zhang, M.; Zhang, W.; and Zhang, M. 2018. SEE: Syntax-aware Entity Embedding for Neural Relation Extraction. In *Proceedings of AAAI*.

Hendrickx, I.; Kim, S. N.; Kozareva, Z.; Nakov, P.; Séaghdha, D. O.; Padó, S.; Pennacchiotti, M.; Romano, L.; and Szpakowicz, S. 2019. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. *arXiv preprint arXiv:1911.10422* .

Katiyar, A.; and Cardie, C. 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)*, 917–928.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Krause, B.; Gotmare, A. D.; McCann, B.; Keskar, N. S.; Joty, S.; Socher, R.; and Rajani, N. F. 2020. Gedi: Generative discriminator guided sequence generation. *arXiv preprint arXiv:2009.06367* .

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural Architectures for Named Entity Recognition. In *HLT-NAACL*.

Li, J.; Wang, R.; Zhang, N.; Zhang, W.; Yang, F.; and Chen, H. 2020. Logic-guided Semantic Representation Learning for Zero-Shot Relation Classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, 2967–2978. Barcelona, Spain (Online): International Committee on Computational Linguistics. URL https://www.aclweb.org/anthology/2020.coling-main.265.

Li, Q.; and Ji, H. 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)*, 402–412.

Lin, Y.; Liu, Z.; and Sun, M. 2017. Neural relation extraction with multi-lingual attention. In *Proceedings of ACL*, volume 1, 34–43.

Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; and Sun, M. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*, volume 1, 2124–2133.

Miwa, M.; and Bansal, M. 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proceedings of ACL*, volume 1, 1105–1116.

Nadeau, D.; and Sekine, S. 2007. A survey of named entity recognition and classification.

Nayak, T.; and Ng, H. T. 2019. Effective Modeling of Encoder-Decoder Architecture for Joint Entity and Relation Extraction. *arXiv preprint arXiv:1911.09886* .

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* .

Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 148–163. Springer.

Roth, D.; and Yih, W.-t. 2007. Global inference for entity and relation identification via a linear programming formulation. *Introduction to statistical relational learning* 553–580.

Schuster, M.; and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing* 45(11): 2673–2681.

Takanobu, R.; Zhang, T.; Liu, J.; and Huang, M. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *In AAAI*, volume 33, 7072–7079.

Wang, Z.; Wen, R.; Chen, X.; Huang, S.-L.; Zhang, N.; and Zheng, Y. 2020. Finding influential instances for distantly supervised relation extraction. *arXiv preprint arXiv:2009.09841* .

Wei, Z.; Jia, Y.; Tian, Y.; Hosseini, M. J.; Steedman, M.; and Chang, Y. 2020a. A Novel Cascade Binary Tagging Framework for Relational Triple Extraction. In *Proceedings of ACL 2020*.

Wei, Z.; Su, J.; Wang, Y.; Tian, Y.; and Chang, Y. 2020b. A Novel Cascade Binary Tagging Framework for Relational Triple Extraction. In *Proceedings of ACL*, 1476–1488.

Wiseman, S.; and Rush, A. M. 2016. Sequence-to-sequence learning as beam-search optimization. *arXiv preprint arXiv:1606.02960* .

Wu, T.; Li, X.; Li, Y.-F.; Haffari, R.; Qi, G.; Zhu, Y.; and Xu, G. 2021. Curriculum-Meta Learning for Order-Robust Continual Relation Extraction. *arXiv preprint arXiv:2101.01926* .

Yonghui, W.; Schuster, M.; Chen, Z.; Le, Q.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. 2016. Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* .

Zeng, D.; Liu, K.; Chen, Y.; and Zhao, J. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of EMNLP*, 1753–1762.

Zeng, D.; Zhang, H.; and Liu, Q. 2020. CopyMTL: Copy Mechanism for Joint Extraction of Entities and Relations with Multi-Task Learning. In *AAAI*, 9507–9514.

Zeng, X.; Zeng, D.; He, S.; Liu, K.; and Zhao, J. 2018a. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of ACL*, 506–514.

Zeng, X.; Zeng, D.; He, S.; Liu, K.; and Zhao, J. 2018b. Extracting Relational Facts by an End-to-End Neural Model with Copy Mechanism. In *Proceedings of ACL*, 506–514. Melbourne, Australia: Association for Computational Linguistics. doi:10.18653/v1/P18-1047. URL https://www.aclweb.org/anthology/P18-1047.

Zhang, N.; Deng, S.; Bi, Z.; Yu, H.; Yang, J.; Chen, M.; Huang, F.; Zhang, W.; and Chen, H. 2020a. OpenUE: An Open Toolkit of Universal Extraction from Text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 1–8. Online. doi:10.18653/v1/2020.emnlp-demos.1.

Zhang, N.; Deng, S.; Li, J.; Chen, X.; Zhang, W.; and Chen, H. 2020b. Summarizing Chinese Medical Answer with Graph Convolution Networks and Question-focused Dual Attention. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 15–24. Online. doi: 10.18653/v1/2020.findings-emnlp.2.

Zhang, N.; Deng, S.; Sun, Z.; Chen, J.; Zhang, W.; and Chen, H. 2020c. Relation Adversarial Network for Low Resource Knowledge Graph Completion. In *Proceedings of The Web Conference 2020*, 1–12.

Zhang, N.; Deng, S.; Sun, Z.; Chen, X.; Zhang, W.; and Chen, H. 2018. Attention-Based Capsule Networks with Dynamic Routing for Relation Extraction. In *Proceedings of EMNLP*.

Zhang, N.; Deng, S.; Sun, Z.; Wang, G.; Chen, X.; Zhang, W.; and Chen, H. 2019a. Long-tail Relation Extraction via Knowledge Graph Embeddings and Graph Convolution Networks. *arXiv preprint arXiv:1903.01306* .

Zhang, S.; Duh, K.; and Van Durme, B. 2017. Selective decoding for cross-lingual open information extraction. In *Proceedings of IJCNLP*, 832–842.

Zhang, Y.; Jiang, Z.; Zhang, T.; Liu, S.; Cao, J.; Liu, K.; Liu, S.; and Zhao, J. 2020d. MIE: A Medical Information Extractor towards Medical Dialogues. In *Proceedings of ACL*, 6460–6469.

Zhang, Y.; Merck, D.; Tsai, E. B.; Manning, C. D.; and Langlotz, C. P. 2019b. Optimizing the factual correctness of a summary: A study of summarizing radiology reports. *arXiv preprint arXiv:1911.02541* .

Zheng, S.; Wang, F.; Bao, H.; Hao, Y.; Zhou, P.; and Xu, B. 2017a. Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. *CoRR* abs/1706.05075. URL http://arxiv.org/abs/1706.05075.

Zheng, S.; Wang, F.; Bao, H.; Hao, Y.; Zhou, P.; and Xu, B. 2017b. Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. In *Proceedings of ACL*, 1227–1236.

Zhu, C.; Hinthorn, W.; Xu, R.; Zeng, Q.; Zeng, M.; Huang, X.; and Jiang, M. 2020. Boosting factual correctness of abstractive summarization with knowledge graph. *arXiv preprint arXiv:2003.08612* .