RelationPrompt: Leveraging Prompts to Generate Synthetic Data for Zero-Shot Relation Triplet Extraction

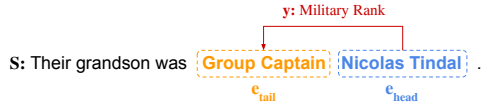
Yew Ken Chia[*1,2] **Lidong Bing**[†1] **Soujanya Poria**[2] **Luo Si**[1]
[1]DAMO Academy, Alibaba Group [2]Singapore University of Technology and Design
{yewken.chia, l.bing, luo.si}@alibaba-inc.com
{yewken_chia, sporia}@sutd.edu.sg

Abstract

Despite the importance of relation extraction in building and representing knowledge, less research is focused on generalizing to unseen relations types. We introduce the task setting of Zero-Shot Relation Triplet Extraction (ZeroRTE) to encourage further research in low-resource relation extraction methods. Given an input sentence, each extracted triplet consists of the head entity, relation label, and tail entity where the relation label is not seen at the training stage. To solve ZeroRTE, we propose to synthesize relation examples by prompting language models to generate structured texts. Concretely, we unify language model prompts and structured text approaches to design a structured prompt template for generating synthetic relation samples when conditioning on relation label prompts (RelationPrompt). To overcome the limitation for extracting multiple relation triplets in a sentence, we design a novel Triplet Search Decoding method. Experiments on FewRel and Wiki-ZSL datasets show the efficacy of RelationPrompt for the ZeroRTE task and zero-shot relation classification. Our code and data are available at github.com/declare-lab/RelationPrompt.

1 Introduction

Relation extraction aims to predict relationships between entities in unstructured text, which has applications such as knowledge graph construction (Lin et al., 2015) and question answering (Xu et al., 2016). However, existing approaches often require large datasets of annotated samples which are costly to annotate and have a fixed set of relations. Currently, less research is focused on the zero-shot setting (Wang et al., 2019) where models need to generalize to unseen relation sets without available annotated samples (Wang et al., 2019).



Task Setting	Input	Output	Supervision
Relation Classification	S, e_{head}, e_{tail}	y	Full
Zero-Shot Relation Classification	S, e_{head}, e_{tail}	y	Zero-Shot
Zero-Shot Relation Slot-Filling	S, e_{head}, y	e_{tail}	Zero-Shot
Relation Triplet Extraction	S	e_{head}, e_{tail}, y	Full
Zero-Shot Relation Triplet Extraction	S	e_{head}, e_{tail}, y	Zero-Shot

Table 1: Comparison of task settings with our proposed Zero-Shot Relation Triplet Extraction (ZeroRTE). To our knowledge, ZeroRTE is the first task to extract full relation triplets in the zero-shot setting.

Although there are existing zero-shot relation task settings, they do not require extracting the full relation triplets. The task setting of Zero-Shot Relation Classification[1] (ZeroRC) was previously introduced by Chen and Li (2021) to classify the relation between a given head and tail entity pair for unseen labels. However, it is not always practical or realistic to assume that the ground-truth entities are readily available. Zero-Shot Relation Slot-Filling (Levy et al., 2017) aims to predict the tail entity based on the provided head entity and relation, but also relies on other methods for entity detection. Thus, it also faces the challenge of error propagation in practice (Zhong and Chen, 2021).

Hence, we propose a new and challenging task setting called Zero-Shot Relation Triplet Extraction (ZeroRTE). The goal of ZeroRTE is to extract triplets of the form (head entity, tail entity, relation label) from each sentence despite not having any annotated training samples that contain the test relation labels. For a clear comparison between task settings, we provide a summary in Table 1. To our knowledge, this is the first work to extend the task of Relation Triplet Extraction to the zero-shot setting. For example in Figure 1, the training samples may belong to the seen relation set {Sibling, Man-

[*]Yew Ken is a student under the Joint PhD Program between Alibaba and SUTD.

[†]Corresponding author.

[1]As relation classification and relation extraction are sometimes interchangeable, we refer to relation classification.

Findings of the Association for Computational Linguistics: ACL 2022, pages 45 - 57

ufacturer, Architect}, while the test samples may belong to the unseen relation set {Military Rank, Position Played, Record Label}. Given the annotated training samples in Figure 1a, ZeroRTE aims to extract triplets such as (Nicolas Tindal, Military Rank, Captain) in Figure 1b.

To solve the challenges of data scarcity, there are several existing approaches. Although distant supervision (Ji et al., 2017) can be used to construct a relation corpus with a many relation types, this approach generally results in lower annotation quality than human annotation. Furthermore, distant supervision remains limited to a fixed set of relation types in the existing knowledge base (Smirnova and Cudré-Mauroux, 2018). Another approach is to formulate the task objective such that the label space is unconstrained. For instance, zero-shot sentence classification can be reframed as entailment (Puri and Catanzaro, 2019) or embedding similarity (Pushp and Srivastava, 2017) objectives. However, the existing formulations are designed for sequence classification tasks, which cannot be directly applied to structured prediction tasks such as relation triplet extraction. A third direction is to leverage pre-trained language models using task-specific prompt templates (Liu et al., 2021) which enables the models to generalize to new tasks with little to no training samples, such as zero-text classification (Zhong et al., 2021). This zero-shot potential is possible by leveraging the semantic information in prompts to query the language comprehension capabilities of pre-trained language models (Radford et al., 2019).

Hence, we propose RelationPrompt which reframes the zero-shot problem as synthetic data generation. The core concept is to leverage the semantics of relation labels, prompting language models to generate synthetic training samples which can express the desired relations. The synthetic data can then be used to train another model to perform the zero-shot task. This capability is supported by the finding that language models can be prompted to control task-specific aspects of the generated text, such as domain and content (Keskar et al., 2019). For instance, given the relation label "Military Rank" in Figure 1c, it is reasonable to condition the language model and compose a sentence demonstrating the relationship that a person has been bestowed with a certain position in the armed forces. Hence, a possible sentence could be "She is the wife of Lieutenant Colonel George Hocham.",

Relation	Sentence
Sibling	She was the mother of Michael and Joel Douglas.
Manufacturer	In late 2012 , Samsung announced its NX300 camera.
Architect	His house was designed by Henry Hob Richardson.

(a) Annotation samples of seen relations for training.

Relation	Sentence
Military Rank	Their grandson was Group Captain Nicolas Tindal.
Position Played	Made Chad Brown the highest paid linebacker in NFL.
Record Label	Deadsy signed onto Immortal Records to release "Phantasmagore".

(b) Annotation samples of unseen relations for evaluation.

Relation	Sentence
Military Rank	She is the wife of Lieutenant Colonel George Hocham.
Position Played	However, it was Dario Argentino who defended the midfield.
Record Label	"The Sun" was first recorded by Pavement in 1982.

(c) Generated synthetic samples of unseen relations.

Figure 1: Example relation triplet data for ZeroRTE and our formulation as synthetic sentence generation. The head and tail entities are shown in blue and orange, respectively. The ZeroRTE train samples (a) and test samples (b) contain triplets that belong to disjoint relation label sets. We formulate ZeroRTE as generating synthetic samples (c) for the unseen test relation labels. The synthetic data can then be used to train another model to extract relation triplets from the test sentences. We also present more data samples in Appendix A.1.

where the head entity is "George Hocham" and the tail entity is "Lieutenant Colonel". Given generated samples of sufficient quality and diversity, the synthetic dataset can effectively supervise another model to perform ZeroRTE.

To encode the relation triplet information as text sequences which can be generated by language models, we unify prompt templates with structured text formats (Paolini et al., 2020). Structured texts use special markers to encode the structured information which can be easily decoded as triplets. However, it is challenging to generate sentences which contain multiple different relation triplets. Designing a complex structured prompt template to encode multiple triplets may compromise the generation quality as the language model needs to manipulate multiple relations at once. Hence, we focus on generating single-triplet samples and explore how this limitation can be overcome by the downstream relation extractor model. Concretely, we propose a method named Triplet Search Decoding which allows the extraction of multiple triplets at prediction time despite training on synthetic samples which contain a single triplet each.

Contributions. In summary, our main contributions include: (1) We introduce the ZeroRTE

task setting which overcomes limitations in prior task settings by extending the Relation Triplet Extraction task to the zero-shot setting. ZeroRTE is released as a publicly available benchmark based on the reorganized FewRel (Han et al., 2018) and Wiki-ZSL (Chen and Li, 2021) datasets. (2) In order to make ZeroRTE solvable in a supervised manner, we propose RelationPrompt to generate synthetic relation examples by prompting language models to generate structured texts. (3) We propose Triplet Search Decoding to overcome the limitation for extracting multiple relation triplets in a sentence. (4) RelationPrompt surpasses prior ZeroRC methods and baselines on ZeroRTE, setting the bar for future work. Our analysis shows that the generated samples are reasonable and diverse, hence serving as effective synthetic training data.

2 RelationPrompt: Methodology

To extract triplets for unseen relation labels in ZeroRTE, we propose a framework called RelationPrompt which uses relation labels as prompts to generate synthetic relation examples of target unseen labels. The synthetic data can then be used to supervise any downstream relation extraction model. Hence, our framework requires two models: a Relation Generator for synthetic relation samples, and a Relation Extractor that will be trained on the synthetic data and used to predict triplets for unseen relations. In order to represent the relation triplet information to be processed by language models, we design structured prompt templates. The relation extractor is designed to support both ZeroRTE and ZeroRC tasks. We further propose Triplet Search Decoding to overcome the challenge of generating relation samples with multiple triplets.

2.1 Task Formulation

In ZeroRTE, the goal is to learn from the seen dataset D_s and generalize to the unseen dataset D_u. The datasets D_s and D_u are used for training and testing respectively, and are originally split from the full dataset which is defined as $D = (S, T, Y)$ where S denotes the input sentences, T denotes the output triplets and Y denotes the set of relation labels present in D. The seen and unseen label sets are predefined and denoted as $Y_s = \{y_s^1, ..., y_s^n\}$ and $Y_u = \{y_u^1, ..., y_u^m\}$ respectively, where $n = |Y_s|$ and $m = |Y_u|$ are the size of seen and unseen label sets respectively. Hence, the label sets of D_s and D_u are disjoint, i.e., $Y_s \cap Y_u = \emptyset$. Each data sample contains the input sentence $s \in S$ which corresponds to a list $t \in T$ which can contain one or more output triplets. A relation triplet is defined as (e_{head}, e_{tail}, y) which denotes the head entity, tail entity and relation label respectively. To solve ZeroRTE, we formulate the following algorithm:

Input	Template	Relation: <Label>.
	Example	Relation: Military Rank.
Output	Template	Context: <Sentence>. Head Entity: <Subject>, Tail Entity: <Object>.
	Example	Context: Their grandson was Captain Nicolas Tindal. Head Entity: Nicolas Tindal, Tail Entity: Captain.

(a) Structured template for relation generator.

Input	Template	Context: <Sentence>.
	Example	Context: Their grandson was Captain Nicolas Tindal.
Output	Template	Head Entity: <Subject>, Tail Entity: <Object>, Relation: <Label>.
	Example	Head Entity: Nicolas Tindal, Tail Entity: Captain, Relation: Military Rank.

(b) Structured template for relation extractor.

Figure 2: RelationPrompt structured templates. The head entities, tail entities and relation labels are shown in blue, orange and dark red respectively. The relation generator (a) takes the relation label as input and outputs the context and entity pair. The relation extractor (b) takes the sentence as input and outputs the relation triplet which consists of entity pair and relation label.

Algorithm 1 RelationPrompt: Prompting language models to generate synthetic data for ZeroRTE.

Define:
Dataset $D = (\text{Sentences } S, \text{Triplets } T, \text{Labels } Y)$

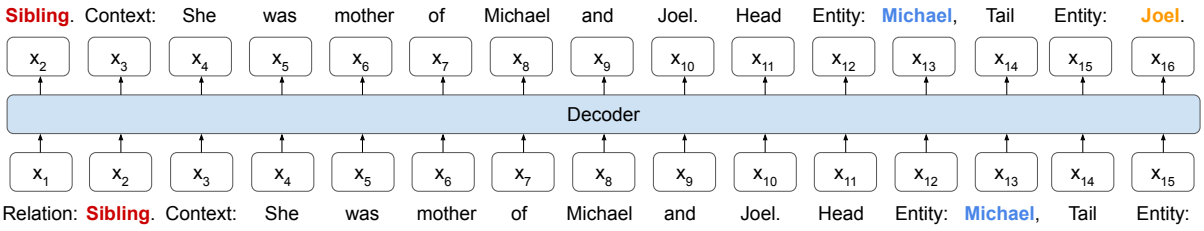
Require: Train Dataset D_s, Test Dataset D_u, Relation Generator M_g, Relation Extractor M_e.

Ensure: $Y_s \cap Y_u = \emptyset$.

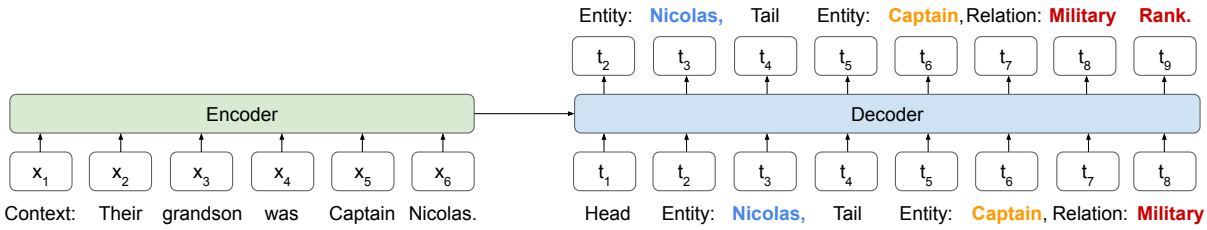
1: $M_{g,finetune} \leftarrow Train(M_g, D_s)$
2: $M_{e,finetune} \leftarrow Train(M_e, D_s)$
3: $D_{synthetic} \leftarrow Generate(M_{g,finetune}, Y_u)$
4: $M_{e,final} \leftarrow Train(M_{e,finetune}, D_{synthetic})$
5: $\hat{T}_u \leftarrow Predict(M_{e,final}, S_u)$
6: **return** Extracted Triplets $\hat{T}_u$

2.2 Relation Generator

Language models are implicitly capable of zero-shot generalization based on their general and large-scale pre-training (Radford et al., 2019). Furthermore, text generation has been shown to be effectively controllable (Keskar et al., 2019). Hence, we prompt the language model to generate synthetic samples by conditioning on the target unseen relation labels. As shown in Algorithm 1, relation generator M_g is first fine-tuned on samples for the seen dataset D_s (line 1) and then prompted by relation labels Y_u to generate the synthetic sam-



(a) Training process for relation generator.



(b) Training process for relation extractor.

Figure 3: Model training process. Each head entity, tail entity and relation label is shown in blue, orange and dark red respectively. To conserve space, the sentences shown are shortened and punctuation is not separated. The relation generator (a) is trained with the standard language modeling objective to condition on the relation label and generate the sentence and entity pair. The relation extractor (b) is trained with the standard sequence-to-sequence objective to condition on the input sentence and output the relation triplet of entity pair and relation label.

ples $D_{synthetic}$ (line 3). As shown in Figure 2a, the relation generator takes as input a structured prompt in the form of "Relation: y" and outputs a structured output in the form of "Context: s. Head Entity: e_{head}, Tail Entity: e_{tail}.". We employ a causal language model as our relation generator to sample the structured sequence in an autoregressive manner. As shown in 3a, the model M_g is trained using the standard language modeling objective of next-word prediction (Bengio et al., 2001). Given each sequence $x = [x_1, x_2, ..., x_n]$, the goal is to learn the conditional generation probability:

$$p(x) = \prod_{i=1}^{n} p(x_i|x_{<i}) \quad (1)$$

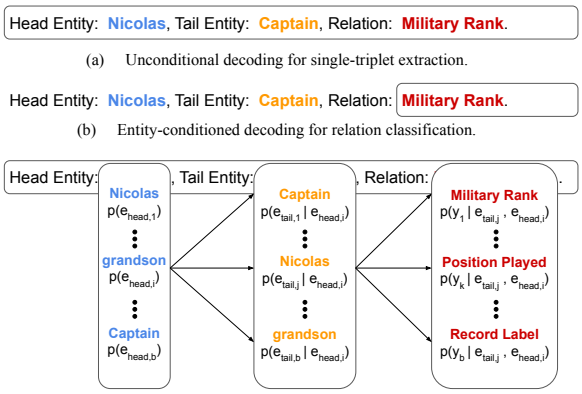
To generate diverse output sequences for each input relation prompt, we use sampling with temperature t (Hinton et al., 2015) over the output logits o and vocabulary size V with temperature tp:

$$p(x_i|x_{<i}) = \frac{exp(o_i/tp)}{\sum_{j=1}^{|V|} exp(o_j/tp)} \quad (2)$$

The output sequences are decoded into relation triplets by splitting on the special terms "Context:", "Head Entity:" and "Tail Entity:". In case of decoding errors where an entity is not found in the generated context, we discard that sample and continue generating until a fixed amount of valid samples is reached.

2.3 Relation Extractor

Given the generated samples of unseen relations, we can train a relation extractor model M_e to predict the relation triplets in a zero-shot setting. As shown in Algorithm 1, relation extractor M_e is first fine-tuned on samples for the seen dataset D_s (line 2) and finally tuned on the synthetic samples $D_{synthetic}$ (line 4). Lastly, M_e is used to predict and extract relation triplets $\hat{T}_u$ from the test sentences S_u (lines 5 and 6). We adopt a sequence-to-sequence learning approach which is flexible and effective for structured prediction tasks (Cui et al., 2021; Paolini et al., 2020). As shown in Figure 2b, the relation extractor takes as input a structured prompt containing the sentence s in the form of "Context: s". It then generates a structured output sequence containing a single pair of entities e_{head} and e_{tail} satisfying the relation y, in the form of "Head Entity: e_{head}, Tail Entity: e_{tail}, Relation: y". As shown in Figure 3b, we use a standard sequence-to-sequence objective (Lewis et al., 2020) for training and greedy decoding for generation. To predict a single relation triplet in a given sentence s, we can generate the model outputs without any initial decoder input, as seen in Figure 4a. In case of invalid entity or relation, we treat it as null prediction for that sample. On the other hand, prediction for ZeroRC is easily supported by providing the entity information as the initial decoder input. As shown



(a) Unconditional decoding for single-triplet extraction.

(b) Entity-conditioned decoding for relation classification.

(c) Triplet search decoding for multi-triplet extraction.

Figure 4: Comparison of generation decoding methods with our proposed Triplet Search Decoding. The head entities, tail entities and relation labels are shown in blue, orange and dark red respectively. Unconditional decoding (a) can be used to predict one relation triplet in each sentence for ZeroRTE. Entity-conditioned decoding (b) can be used to predict only the relation label between the given entity pair for ZeroRC. Our proposed triplet search decoding (c) can be used to predict multiple triplets in each sentence for ZeroRTE.

in Figure 4b, the model takes "Context: s, Head Entity: e_{head}, Tail Entity: e_{tail}, Relation:" as decoder input to generate "y" as output. Hence, our method naturally encompasses both ZeroRTE and ZeroRC as this change affects model prediction and not training.

2.4 Extracting Multiple triplets using Triplet Search Decoding

We further propose a generation decoding method in order to improve the zero-shot extraction performance on sentences which contain multiple triplets. For the RelationPrompt generation of synthetic data, each sample is limited to contain a single relation triplet. Hence, conventional models for triplet extraction most likely cannot perform well with our framework for multi-triplet ZeroRTE as they normally assume that the training samples may contain multiple triplets per sentence. The inference method of multi-turn question answering (Li et al., 2019) may mitigate this issue, but cannot scale easily to unseen relations as it relies on hand-crafted question templates which are specific to certain relation and entity types. Hence, we propose Triplet Search Decoding which improves multi-triplet ZeroRTE for the relation extractor.

Given the relation extractor which takes a sentence as input and generates output sequences in an autoregressive fashion, greedy decoding as in Figure 4a can output a single sequence. However, Triplet Search Decoding as shown in Figure 4c can output multiple sequences that each correspond to a different candidate relation triplet. We then apply a likelihood threshold to filter the final output sequences. The core concept is enumerating multiple output sequences during generation by considering multiple candidates for the head entity, tail entity and relation label respectively. Starting from the special sub-sequence "Head Entity:", it follows from our template in Figure 3b that the next generated token should be the first token of the head entity, such as "Nicolas". For the i^{th} possible first token of the head entity, we denote the softmax probability as $p(e_{head,i})$. We only consider the probability of the first token as it can mostly determine the following generated tokens of the entity (Zhao et al., 2021). Instead of greedily decoding the entire sequence, we branch the generation into b sequences based on the tokens with the top b highest $p(e_{head,i})$. Thereafter, the sequence is greedily decoded until the special sub-sequence "Tail Entity:" is generated. The following token will then be the first token of the tail entity, such as "Captain". The j^{th} tail entity first token probability is denoted as $p(e_{tail,j}|e_{head,i})$. Hence, the generation is branched such that for each head entity, there will be another b sequences based on the tokens with the top b highest $p(e_{tail,j}|e_{head,i})$. Thereafter, the sequence is greedily decoded until the special sub-sequence "Relation:" is generated. The next generated token will be the first token of the relation label, such as "Military" in "Military Rank". The k^{th} relation first token probability is denoted as $p(y_k|e_{head,i}, e_{tail,j})$. We branch the generation such that for each pair of head entity and tail entity, there will be another b sequences based on the tokens with the top b highest $p(y_k|e_{head,i}, e_{tail,j})$. For each sequence, the generation proceeds greedily until the end token is reached, and the overall inference probability is aggregated as:

$$\begin{aligned} p(triplet_{i,j,k}) &= p(e_{head,i}, e_{tail,j}, y_k) \\ &= p(y_k|e_{head,i}, e_{tail,j}) \\ &\quad \cdot p(e_{tail,j}|e_{head,i}) \\ &\quad \cdot p(e_{head,i}) \end{aligned} \tag{3}$$

We note that the conditional assumption does not directly consider the other context tokens as they consist of the special sub-sequences which are fixed as part of our generation template. The input sen-

tence s is also not included in the formulation as it is the same when considering multiple output triplets for one sample. At this point, there will be b^3 sequences, each corresponding to a different candidate relation triplet. To filter the final output sequences, we use a probability threshold over that is tuned on the validation F_1 metric, with hyperparameter details in Section A.2. Compared to previous generative extraction methods (Paolini et al., 2020; Nayak and Ng, 2020), Triplet Search Decoding allows the probability $p(triplet_{i,j,k})$ of each output triplet to be directly calculated and hence control the number of output triplets using the threshold. Compared to other decoding methods such as beam search, Triplet Search Decoding leverages the specific relation triplet structure in our structured text templates. Hence, it can ensure that each output sequence corresponds to a different relation triplet. Furthermore, Triplet Search Decoding is more interpretable than existing generative methods for triplet extraction as it can directly provide the prediction probability for each triplet. More importantly for ZeroRTE, this decoding process allows the relation extractor to naturally predict multiple triplets at test time despite training on synthetic samples which have a single triplet each.

3 Experiments

3.1 Datasets

We use the following two datasets for our experiments. FewRel (Han et al., 2018) was hand-annotated for few-shot relation extraction, but we made it suitable for the zero-shot setting after data splitting into disjoint relation label sets for training, validation and testing. Wiki-ZSL (Chen and Li, 2021) is constructed through distant supervision over Wikipedia articles and the Wikidata knowledge base. The dataset statistics are shown in Table 2. To partition the data into seen and unseen label sets, we follow the same process as Chen and Li (2021) to be consistent. For each dataset, a fixed number of labels are randomly selected as unseen labels while the remaining labels are treated as seen labels during training. To study the performance of methods under different settings of unseen label set size m, we use $m \in \{5, 10, 15\}$ in our experiments. In order to reduce the effect of experimental noise, the label selection process is repeated for five different random seeds to produce different data folds. For each data fold, the test set consists of the sentences containing unseen labels. Five validation labels from the seen labels are used to select sentences for early stopping and hyperparameter tuning. The remaining sentences are treated as the train set. Hence, the zero-shot setting ensures that train, validation and test sentences belong to disjoint label sets.

	Samples	Entities	Relations	Sentence Length
Wiki-ZSL	94,383	77,623	113	24.85
FewRel	56,000	72,954	80	24.95

Table 2: Dataset statistics. "Sentence Length" refers to the average number of words in each sentence.

3.2 Experimental Settings

For the relation generator, we fine-tune the pre-trained GPT-2 (Radford et al., 2019) which has 124M parameters. For the relation extractor, we fine-tune the pre-trained BART (Lewis et al., 2020) which has 140M parameters. In both cases, the fine-tuning is performed on the training set for up to five epochs and early stopping is based on the validation loss. The learning rate is 3e-5 with linear warm up for the first 20% of training steps and batch size is set to 128. During the training process, we use the AdamW optimizer (Loshchilov and Hutter, 2019). The relation generator is used to generate synthetic samples based on the validation and test set label names. A fixed amount of sentences will be generated for each relation. The relation extractor is fine-tuned again on the synthetic relation sentences and then used for evaluation on the test set.[2]

To perform evaluation for ZeroRTE, we evaluate the triplet extraction results separately for sentences containing single triplets and multiple triplets. To evaluate multiple triplet extraction, we use the Micro F_1 metric which is standard in structured prediction tasks (Paolini et al., 2020) and report the precision (P.) and recall (R.). Evaluating single triplet extraction involves only one possible triplet for each sentence, hence the metric used is Accuracy (Acc.). We evaluate on ZeroRC using the Macro F_1 metric to be consistent with Chen and Li (2021). Table 3 and 4 report the average results across five data folds as detailed in Section 3.1.

3.3 Baseline Methods

ZeroRTE As ZeroRTE is a new task setting, we provide two baseline methods for comparison with our RelationPrompt method. Firstly, our relation

[2]See Appendix A.2 for more implementation details.

Unseen Labels	Model	Single Triplet		Multi Triplet					
		Wiki-ZSL	FewRel	Wiki-ZSL			FewRel		
		Acc.	*Acc.*	*P.*	*R.*	F_1	*P.*	*R.*	F_1
m=5	TableSequence (Wang and Lu, 2020)	14.47	11.82	**43.68**	3.51	6.29	15.23	1.91	3.40
	NoGen	9.05	11.49	15.58	**43.23**	22.26	9.45	**36.74**	14.57
	RelationPrompt	**16.64**	**22.27**	29.11	31.00	**30.01**	**20.80**	24.32	**22.34**
m=10	TableSequence (Wang and Lu, 2020)	9.61	12.54	**45.31**	3.57	6.4	**28.93**	3.60	6.37
	NoGen	7.10	12.40	9.63	**45.01**	15.70	6.40	**41.70**	11.02
	RelationPrompt	**16.48**	**23.18**	30.20	32.31	**31.19**	21.59	28.68	**24.61**
m=15	TableSequence (Wang and Lu, 2020)	9.20	11.65	**44.43**	3.53	6.39	**19.03**	1.99	3.48
	NoGen	6.61	10.93	7.25	**44.68**	12.34	4.61	**36.39**	8.15
	RelationPrompt	**16.16**	**18.97**	26.19	32.12	**28.85**	17.73	23.20	**20.08**

Table 3: Results for Zero-Shot Relation Triplet Extraction (ZeroRTE).

extractor can be made to perform ZeroRTE without fine-tuning on synthetic samples as it is trained to extract triplets on the sentences of the seen relation set. At prediction time, we constrain the generated labels to be selected from the target label names by masking the generated token probabilities. We denote this model as **"NoGen"** to indicate that it does not use generated synthetic samples for training. Secondly, we use an existing triplet extraction model known as **TableSequence** (Wang and Lu, 2020). As it is normally unable to perform ZeroRTE, we provide supervision using synthetic samples from our relation generator.

ZeroRC There are three main categories of competing methods for ZeroRC. Firstly, **R-BERT** (Wu and He, 2019) is a relation classification model but can be adapted to the zero-shot setting by using the sentence representations to perform nearest neighbor search over label embeddings. Next, **CIM** (Rocktäschel et al., 2016) is an entailment-based method which takes the sentence and each possible relation as input to perform binary classification whether the label matches the sentence semantically. Lastly, **ZS-BERT** (Chen and Li, 2021) generates sentence representations that are conditioned on the provided entity pair information, and performs nearest neighbor search over embeddings of the candidate relation descriptions.

3.4 Experimental Results

Triplet Extraction We compare RelationPrompt with the baselines on ZeroRTE for Wiki-ZSL and FewRel datasets in Table 3. In both single-triplet and multi-triplet evaluation, our method consistently outperforms the baseline methods in terms of Accuracy and F_1 metrics respectively. Although we do not observe a consistent advantage in precision and recall scores for multi-triplet extraction, the baseline methods cannot achieve a balanced precision-recall ratio, leading to poor overall F_1 results. The results difference between NoGen and RelationPrompt also indicate that using the synthetic samples from the relation generator is critical, as the F_1 score can be improved by more than two times in some cases. This also suggests that the relation generator can produce reasonable-quality synthetic sentences as training data for the downstream relation extractor. We also observe that the choice of relation extractor for ZeroRTE is not trivial, as the third-party TableSequence (Wang and Lu, 2020) has significantly worse performance when compared to RelationPrompt, especially for multi-triplet extraction. Although the TableSequence model is able to perform multi-triplet extraction by design, it assumes that the training data may contain multi-triplet sentences, whereas our synthetic data is limited to single triplet samples. On the other hand, our proposed relation extractor and decoding method effectively overcomes this chal-

Unseen Labels	Model	Wiki-ZSL			FewRel		
		P.	*R.*	F_1	*P.*	*R.*	F_1
m=5	R-BERT	39.22	43.27	41.15	42.19	48.61	45.17
	CIM	49.63	48.81	49.22	58.05	61.92	59.92
	ZS-BERT	**71.54**	72.39	71.96	76.96	78.86	77.90
	NoGen	51.78	46.76	48.93	72.36	58.61	64.57
	RelationPrompt	70.66	**83.75**	**76.63**	**90.15**	**88.50**	**89.30**
m=10	R-BERT	26.18	29.69	27.82	25.52	33.02	28.20
	CIM	46.54	47.90	45.57	47.39	49.11	48.23
	ZS-BERT	60.51	60.98	60.74	56.92	57.59	57.25
	NoGen	54.87	36.52	43.80	66.47	48.28	55.61
	RelationPrompt	**68.51**	**74.76**	**71.50**	**80.33**	**79.62**	**79.96**
m=15	R-BERT	17.31	18.82	18.03	16.95	19.37	18.08
	CIM	29.17	30.58	29.86	31.83	33.06	32.43
	ZS-BERT	34.12	34.38	34.25	35.54	38.19	36.82
	NoGen	54.45	29.43	37.45	66.49	40.05	49.38
	RelationPrompt	**63.69**	**67.93**	**65.74**	**74.33**	**72.51**	**73.40**

Table 4: Zero-Shot Relation Classification (ZeroRC).

Model	F_1	ΔF_1
Full Method	**28.41**	
– Triplet Search Decoding	14.53	-13.88
– Extractor Fine-Tuning (Seen Relations)	13.57	-14.84

Table 5: Ablation results for multi-triplet ZeroRTE.

lenge by naturally enumerating and ranking multiple triplets at inference time.

Relation Classification RelationPrompt naturally supports the ZeroRC task without additional training by providing the entity pair information in the prompt. In Table 4, we observe consistent improvements compared to the prior state-of-the-art method ZS-BERT (Chen and Li, 2021). Notably, our method is able to maintain a relatively high classification F_1 performance when the unseen label set size m increases, whereas ZS-BERT shows a sharper drop in performance. The trend suggests that RelationPrompt is able to scale better to larger unseen label sets, which is more important for open-domain applications. This advantage may further indicate that our method can leverage the semantic information of relation labels more effectively through the token-level conditional generation and extraction stages. On the other hand, ZS-BERT relies on sequence-level representations which can only preserve the high-level label semantics.

4 Analysis

4.1 Ablation Study

We conduct an ablation study to examine the performance of our decoding method and task-specific fine-tuning on the seen relation set for multi-triplet ZeroRTE, and the results are shown in Table 5. The comparison is conducted on the Wiki-ZSL validation set with 10 unseen labels. The large performance gap shows that Triplet Search Decoding is critical for multi-triplet ZeroRTE, and suggests that the enumeration and ranking of relation triplet candidates are of sufficiently high quality. Secondly, we observe a significant drop in performance when the relation extractor is not fine-tuned on seen relation samples from the train set before the final tuning on generated synthetic samples for unseen labels. This case suggests that the initial fine-tuning on sentences for seen relations is useful for learning the general task of relation triplet extraction. The learned representations can then be further fine-tuned on the synthetic samples to adapt specifically for the unseen relations to achieve optimal results.

4.2 Effect of Generated Data Size

We further study how the number of generated synthetic samples effects the multi-triplet ZeroRTE performance. The evaluation is based on Wiki-ZSL validation set with 10 unseen labels, and the results are shown in Figure 6. Increasing the amount from 125 to 250 samples per label improves F_1 score. However, further increasing the generated size up to 2000 does not improve the final performance. This indicates that although the synthetic data is beneficial for ZeroRTE, excessive amounts can lead to over-fitting due to noise. We further analyze the generation diversity in Appendix A.3.

4.3 Qualitative Analysis

To assess how the relation data generator generalizes to relations in the wild, we present several samples of real and generated samples in Figure 5. The relation labels and real sentences were collected from factual articles. Given the relations "Investor", "Defeated By" and "Currency Of", the generator is able to determine the correct semantic meaning of the relations and compose reasonable sentences. In most cases, the generated head and tail entity pairings can match the given relations and have a similar context to the real sentences. However, in the last case for relation "Political Partner", the generated entity pair does not match the relation meaning despite being grounded in a political context. Instead, the generated sentence expresses a relationship that is closer to "Political Party". This suggests that a future area of improvement could be to match the generated head and tail entity more closely to the given relation.

5 Related Work

Zero-Shot Relation Extraction Zero-shot relation extraction was previously framed as a slot-filling task and solved by reading comprehension methods (Levy et al., 2017). However, their approach requires manual template design for each relation label, which cannot scale well to new relation types. Another approach to zero-shot relation extraction is the formulation into an entailment task (Obamuyide and Vlachos, 2018), which is not constrained to a fixed relation label space. Instead, the entailment approach determines if arbitrary pairs of sentences and relation labels are compatible. However, it is designed for sentence classification and cannot be applied to ZeroRTE.

Relation	Real Sentence	Generated Sentence
Investor	In October 1999, Alibaba received a US $25 million investment from Goldman Sachs and SoftBank.	In 2001, CSC announced that it is investing in a record 65 plants in South Africa and plans to invest $2 billion in hydropower, hydrological services and agriculture.
Defeated By	National shuttler Loh Kean Yew recorded a 24-22, 21-14 win over All-England champion Lee Zii Jia.	The team lost three consecutive draws to the New York Knicks, including a 9–2 win against the Lakers
Currency Of	El Salvador, on the other hand, has accepted Bitcoin (BTC) as a legal currency.	In 2001, as the euro strengthened, Italy introduced an extension of the Eurobill (Eurobills), the currency of the Eurozone.
Political Partner	Beijing and the Holy See reached a provisional agreement in 2018 on the appointment of Chinese bishops.	His son, Thomas, was a leading Republican, elected to the Massachusetts State Senate in 1881.

Figure 5: Case study between real and generated samples for relations in the wild. The head and tail entities are shown in blue and orange respectively.

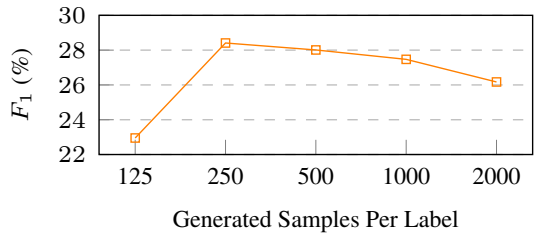


Figure 6: Effect of generated data size on ZeroRTE.

Data Augmentation A popular method for improving model performance in supervised low-resource tasks is data augmentation. Simple heuristics such as token manipulation (Kobayashi, 2018) were initially developed, new methods in language modeling improved the quality of augmented samples (Xie et al., 2020; Wei and Zou, 2019). Although there are data augmentation methods that can be applied to structured tasks such as named entity recognition (Ding et al., 2020) and relation extraction (Papanikolaou and Pierleoni, 2020; Lee et al., 2021), they require existing training samples and cannot be easily adapted to zero-shot tasks.

Knowledge Retrieval RelationPrompt also leverages the knowledge stored in language models (Roberts et al., 2020) to compose relation samples that are grounded in realistic contexts. To ensure that the generated samples are factually accurate, the language model requires strong knowledge retrieval capabilities (Petroni et al., 2019).

Language Model Prompts Prompting-based methods have shown promise as a new paradigm for zero-shot or few-shot inference in natural language processing (Liu et al., 2021). Another advantage is the potential to adapt very large language models (Reynolds and McDonell, 2021) to new tasks without relatively expensive fine-tuning. Concurrent works (Meng et al., 2022; Ye et al., 2022) also show that language models can generate synthetic training data. However, such methods have not yet proven effective for more complex tasks such as triplet extraction.

Structured Prediction RelationPrompt generates synthetic data for relation triplet extraction, which is a structured prediction task. Hence, it can be widely applicable to other structured prediction tasks such as named entity recognition (Aly et al., 2021), event extraction (Huang et al., 2018) or aspect sentiment triplet extraction (Xu et al., 2021).

6 Conclusions and Future Work

In this work, we introduce the task setting of Zero-Shot Relation Triplet Extraction (ZeroRTE) to overcome fundamental limitations in previous task settings and encourage further research in low-resource relation extraction. To solve ZeroRTE, we propose RelationPrompt and show that language models can effectively generate synthetic training data through relation label prompts to output structured texts. To overcome the limitation for extracting multiple relation triplets in a sentence, we propose the Triplet Search Decoding method which is effective and interpretable. Results show that our method surpasses prior ZeroRC methods as well as strong baselines on ZeroRTE, setting the bar for future work. As mentioned in Section 4.3, a future direction for improvement could be to ensure that the generated entity spans are more compatible with the semantics of the relation in the language model prompt.

References

Rami Aly, Andreas Vlachos, and Ryan McDonald. 2021. Leveraging type descriptions for zero-shot named entity recognition and classification. In *Proc. of ACL*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2001. A neural probabilistic language model. In *Proc. of NeurIPS*.

Chih-Yao Chen and Cheng-Te Li. 2021. Zs-bert: Towards zero-shot relation extraction with attribute representation learning. In *Proc. of NAACL*.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of ACL-IJCNLP*.

Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. 2020. DAGA: Data augmentation with a generation approach for low-resource tagging tasks. In *Proc. of EMNLP*.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proc. of EMNLP*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *Proc. of ACL*.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2017. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *Proc. of AAAI*.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *CoRR*, arXiv:1909.05858.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proc. of NAACL*.

Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. 2021. Neural data augmentation via example extrapolation. *CoRR*, arXiv:2102.01335.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proc. of CoNLL*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proc. of ACL*.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *Proc. of ACL*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proc. of AAAI*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, arXiv:2107.13586.

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

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *CoRR*, arXiv:2202.04538.

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *Proc. of AAAI*.

Abiola Obamuyide and Andreas Vlachos. 2018. Zero-shot relation classification as textual entailment. In *Proc. of FEVER*.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2020. Structured prediction as translation between augmented natural languages. In *Proc. of ICLR*.

Yannis Papanikolaou and Andrea Pierleoni. 2020. DARE: data augmented relation extraction with GPT-2. *CoRR*, arXiv:2004.13845.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proc. of EMNLP-IJCNLP*.

Raul Puri and Bryan Catanzaro. 2019. Zero-shot text classification with generative language models. *CoRR*, arXiv:1912.10165.

Pushpankar Kumar Pushp and Muktabh Mayank Srivastava. 2017. Train once, test anywhere: Zero-shot learning for text classification. *CoRR*, arXiv:1712.05972.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI*.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Proc. of CHI*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proc. of EMNLP*.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *Proc. of ICLR*.

Alisa Smirnova and Philippe Cudré-Mauroux. 2018. Relation extraction using distant supervision: A survey. *CSUR*, 51(5):1:35.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proc. of EMNLP*.

Wei Wang, Vincent W. Zheng, Han Yu, and Chunyan Miao. 2019. A survey of zero-shot learning: Settings, methods, and applications. *TIST*, 10(2):1:37.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proc. of EMNLP-IJCNLP*.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proc. of CIKM*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. In *Proc. of NeurIPS*.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on freebase via relation extraction and textual evidence. In *Proc. of ACL*.

Lu Xu, Yew Ken Chia, and Lidong Bing. 2021. Learning span-level interactions for aspect sentiment triplet extraction. In *Proc. of ACL*.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. *CoRR*, arXiv:2202.07922.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proc. of ICML*.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Meta-tuning language models to answer prompts better. In *Findings of EMNLP*.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proc. of NAACL*.

Relation	Sentence
Mouth of Watercourse	It drains into the Pacific Ocean via the Patía River.
Position Played	Made Chad Brown the highest paid linebacker in NFL history.
League	The Diamondbacks compete in the National League West division.
Military Branch	The 47th Liaison Squadron is an inactive United States Air Force unit.
Head of Government	Following the September 2014 general elections in Montserrat, Reuben Meade's government was replaced by new government led by Donaldson Romeo.
Director	The Locket is a 1946 film directed by John Brahm.
Military Rank	General Sir Bernard Paget died on 16 February 1961.
Residence	Diederik van Dijk is married and lives in Benthuizen.
Location	He gave the Bampton Lectures at Oxford in 1824.
Original Language	Her latest Tamil film was "Jaihind 2".

(a) Annotation Samples of Unseen Relations in FewRel Dataset

Relation	Sentence
Employer	Martha Crago is Vice President of Research at Dalhousie University.
Award Received	Private Bernard McQuirt won the Victoria Cross at Rowa.
Sports Discipline	Andrii Toptun is a Ukrainian marathon runner.
Spouse	Messalina, Roman wife of Claudius.
Country of Citizenship	Jarmo Saari is Finnish a guitarist , composer and producer .
Part Of	Line 2 of Metro Bilbao starts at Basauri and reaches Santurtzi.
Official Language	Mass media in Israel in a language other than Hebrew.
Drafted By	Sihugo Green from Duquesne University was selected first overall by the Rochester Royals.
Narrative Location	Aimée & Jaguar is a 1999 German drama film set in Berlin during World War II.
Educated At	Roger Morris earned his doctorate in government from Harvard University.

(b) Annotation Samples of Unseen Relations in Wiki-ZSL Dataset

Figure 7: Additional sentence samples from the datasets. The head and tail entities are shown in blue and orange, respectively.

Relation	Sentence
Mouth of Watercourse	The Cascades River is a freshwater estuary in Florida.
Position Played	In 2009, Wojciech Szczerbiński was named head coach.
League	The 2014 FIFA World Cup, played at Düsseldorf stadium.
Military Branch	At this time the Army continued to deploy to Somalia.
Head of Government	The Prime Minister is the Prime Minister of Pakistan.
Director	"Téléchargier" was directed by the director Olivier Delpierre.
Military Rank	He was a former admiral named Thomas J. Tarr.
Residence	Toretto was born and raised in Nieuwland, Norway.
Location	The district was originally assigned to the Northern Romanovs of Moscow.
Original Language	It was also written by the Finnish filmmaker Mikael Njoro.

(a) Generated Samples of Unseen Relations in FewRel Dataset

Relation	Sentence
Employer	Bewley was signed into the HGV at the age of 17.
Award Received	In 1962 he won Best Director for Unrequited Love.
Sports Discipline	Thomas Stuestor was a champion of tennis in 1872.
Spouse	It was created for Harry M. Truman's wife Nancy in 1950.
Country of Citizenship	Peter Paul Rubens was a Czechoslovak politician and businessman.
Part Of	The main source of power in the Middle East was Saudi Arabia and Egypt.
Official Language	The first official English translation is by Robert Knecht.
Drafted By	In addition, the Cincinnati Bengals drafted quarterback Danny Franklin.
Narrative Location	A story from the English drama series The Tudors.
Educated At	Tchaikov attended the Krasnoyarsk Academy (1960s) in Moscow.

(b) Generated Samples of Unseen Relations in Wiki-ZSL Dataset

Figure 8: Additional synthetic samples from the generated outputs. The head and tail entities are shown in blue and orange, respectively.

A Appendix

A.1 Additional Data Samples

Dataset Samples To further illustrate the datasets used, we show test samples in Figure 7. The samples are taken from the FewRel (a) and Wiki-ZSL (b) test sets respectively with 10 unseen relation labels.

Synthetic Samples To further examine the output of the relation generator, we show test samples in Figure 8. The samples are generated from the FewRel (a) and Wiki-ZSL (b) test set labels respectively with 10 unseen relation labels.

A.2 Implementation Details

Generating Structured Texts We use the relation generator model to generate synthetic sentences in an autoregressive fashion. To convert the structured text outputs to relation triplet samples, we perform simple string processing on the output templates shown in Figure 3a to separate the structured content from the natural text. In case of a small amount of conversion errors, we continue to generate samples until the amount of sentences generated per label is reached. For the relation extractor model, we perform a similar processing on the output templates in Figure 3b to extract the predicted relation triplets. However, in case of processing errors, we do not continue generation and instead treat it as a prediction failure for that input sample.

Hyperparameters We show more detailed hyperparameters used in Table 6. We run a grid search on the Wiki-ZSL validation set with 10 unseen labels for multi-triplet ZeroRTE F_1 metric. A grid search is used to tune the hyperparameters. For number of generated samples per label, we consider the values $\{125, 250, 500, 1000, 2000\}$. To tune the Triplet Search Decoding threshold, we consider fifty evenly-spaced values from the interval over the minimum and maximum output scores of all candidate triplets on the validation set. Due to computational constraints, we consider the number of branches to consider at each stage a fixed value, and do not tune it as a hyperparameter.

Computing Infrastructure The experiments are conducted on NVIDIA V100 GPUs, and each experiment is run on a single GPU with 32 GB of

	Value
Generator Maximum Sequence Length	128
Generator Sampling Top-K	50
Generator Sampling Temperature	1.0
Extractor Maximum Input Length	128
Extractor Maximum Output Length	128
Training Dropout Probability	0.1
Generated Samples Per Label	250
Triplet Search Decoding Top-N Branches	4
Triplet Search Decoding Threshold	-0.9906

Table 6: Additional hyperparameters.

	Samples	Unique Entities	Unique Words
Real Data	3461	3090	14736
Generated Data	3461	4949	10558

Table 7: Data diversity comparison.

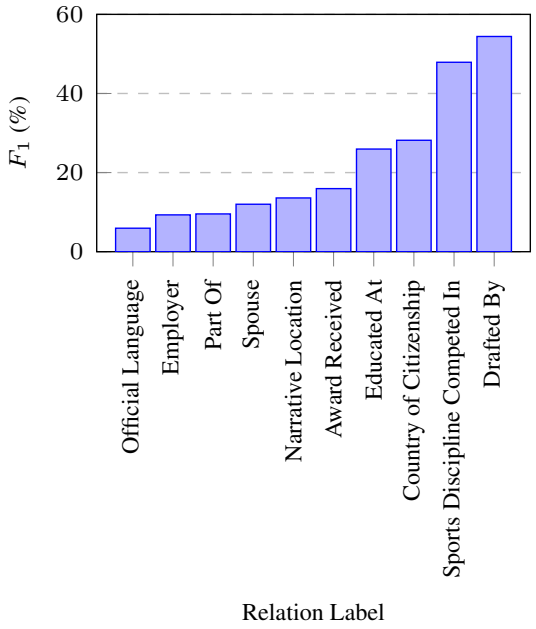


Figure 9: Separate evaluation on relation labels.

memory and mixed precision settings.

A.3 Further Analysis

Generated Sample Diversity Our method for ZeroRTE heavily depends on the quality of the generated data. Hence, we compare the diversity of real and synthetic data samples. Concretely, we measure the number of unique words and entities present in the texts. We used the Wiki-ZSL validation set sentences with five unique labels and generate an equal amount of synthetic sentences for comparison. Table 7 shows that the diversity of unique entities is actually greater for the generated sentences. However, the generated sentences have lower diversity of overall unique words. This may be explained by the fact that entity names tend to be unique, and the generator language model has seen a vast number of unique entity names during the large-scale pre-training. On the other hand, the total unique words are mostly determined by the non-entity words. By using prompts to condition the generation of sentences specifically for unseen relation labels, this may constrain the diversity of contextual information in the output sentences.

Performance Across Relations To study how the performance varies across different relation labels, we evaluate single-triplet ZeroRTE on the Wiki-ZSL test set with 10 unseen labels. Figure 9 shows that the model is able to perform well for relations such as "Drafted By" and "Sports Discipline Competed In". However, it performs more poorly for relations such as "Official Language" and "Employer". This suggests that RelationPrompt performs best for relations which are highly specific to constrain the output context more effectively.