Zero-shot Triplet Extraction by Template Infilling

Bosung Kim[Ψ*]   **Hayate Iso**[℮]   **Nikita Bhutani**[℮]
Estevam Hruschka[℮]   **Ndapa Nakashole**[Ψ]   **Tom Mitchell**[℮,ℛ]
[Ψ]University of California, San Diego   [℮]Megagon Labs   [ℛ]Carnegie Mellon University
bosungkim@ucsd.edu   nnakashole@eng.ucsd.edu
{hayate,nikita,estevam,tom}@megagon.ai

Abstract

The task of triplet extraction aims to extract pairs of entities and their corresponding relations from unstructured text. Most existing methods train an extraction model on training data involving specific target relations, and are incapable of extracting new relations that were not observed at training time. Generalizing the model to unseen relations typically requires fine-tuning on synthetic training data which is often noisy and unreliable. We show that by reducing triplet extraction to a template infilling task over a pre-trained language model (LM), we can equip the extraction model with zero-shot learning capabilities and eliminate the need for additional training data. We propose a novel framework, ZETT (**ZE**ro-shot **T**riplet extraction by **T**emplate infilling), that aligns the task objective to the pre-training objective of generative transformers to generalize to unseen relations. Experiments on FewRel and Wiki-ZSL datasets demonstrate that ZETT shows consistent and stable performance, outperforming previous state-of-the-art methods, even when using automatically generated templates.[1]

1 Introduction

Extracting pairs of entities and their relations from unstructured text is vital to several applications including knowledge base population, text retrieval and question answering (Lin et al., 2015; Xu et al., 2016). Traditional approaches obtain entity pairs and relations step-by-step by considering entity recognition and relation classification as two separate sub-tasks. However, such multi-step approaches suffer from cascading errors and ignore interdependence between the tasks. Recent studies aim at extracting entities and relations together in a single step (Li and Ji, 2014; Zheng et al., 2017;

c: Ay Juancito is an Argentine drama film directed by Héctor Olivera.

Relations: *director_of, lyrics_by, drafted_by, founded_by, located_in*

c: Ay Juancito is an Argentine drama film directed by Héctor Olivera.

Triplet: *(Héctor Olivera, director_of, Ay Juancito)*
        h r t

Figure 1: Triplet extraction task: given a context c and a set of relations $\mathcal{R}$, extract pairs of entities and their relations. Zero-shot extraction aims at extracting relations not covered in the training data.

Paolini et al., 2021). Given a set of pre-defined relations and an input text, they extract triplets of the form (head, relation, tail). We refer to this task as *triplet extraction* (illustrated in Figure 1).

If the relations are pre-defined, an extraction model can be trained on large-scale labeled data acquired via distant supervision or crowdsourcing (Sorokin and Gurevych, 2017; Han et al., 2018). However, such methods are hard to adopt in real-world scenarios where ground-truth entities and relations cannot be specified in advance. To overcome these limitations, there is an increasing interest in generalizing models to extract entities and relations that are not observed during training — a zero-shot setting.

Automatically generating training data for unseen relations is a widely used approach to render zero-shot capabilities to an extraction model. Distant supervision (Mintz et al., 2009; Zeng et al., 2015; Ji et al., 2017) and data augmentation (Chia et al., 2022) can provide automatically labeled training data, but they suffer from quality and consistency of the synthetic data. They also require the model to be further fine-tuned on the synthetic data which can be computationally intensive. Yet another set of approaches extract unseen relations using cross-task knowledge learned from a collection of similar datasets and tasks (Zhong et al., 2021; Sanh et al., 2022) . However, the performance of these methods depends on the similarity between

---

the datasets or the tasks.

Recent progress in large language models (LMs) such as GPT-3 shows that they can be adapted to zero-shot settings if the task objective is aligned with the pre-training objective (Brown et al., 2020). Based on this idea, many NLP tasks such as text classification (Zhong et al., 2021) and relation classification (Levy et al., 2017; Obamuyide and Vlachos, 2018) have been successfully reformulated into prompt-based tasks. However, the state-of-the-art zero-shot triplet extraction approach still relies on synthetic data to generalize to unseen relations (Chia et al., 2022). Moreover, most existing prompting approaches are designed for simple classification or generation tasks, making them unsuitable for the structured prediction such as triplet extraction which requires the identification of the complex triplet format. This work is the first study to explore how to reformulate triplet extraction into a prompt-based method to effectively and efficiently generalize to unseen relations.

We formulate triplet extraction as a template infilling task and propose a novel framework, ZETT (**ZE**ro-shot **T**riplet extraction by **T**emplate infilling) based on an end-to-end generative language model, T5 (Raffel et al., 2019). Concretely, ZETT extends the input text with a relation template (e.g., "<X> is nominated for <Y>") and learns to generate the correct entity pair (e.g., "<X> John Bright <Y> Best Story") for the relation. In this manner, it aligns the task objective with the pre-training objective of the T5 model. The model is fine-tuned using annotated examples and a relation template for each relation in a set of predefined relations. Then at inference, the model extends the input text with template for each unseen relation and generates an entity pair and its score. It uses these scores to rank the relations and entity pairs and output the most-likely triplet(s).

Although the model relies on a template for each seen and unseen relation, we show that ZETT can perform well even with automatically-generated templates for the relations. We further propose optimizations based on relation descriptions to improve efficiency at inference. Figure 2 shows the overview of ZETT. Note that ZETT adopts an efficient single-step approach for the task that does not require synthetic data or additional fine-tuning for unseen relations.

Experiments on publicly available datasets FewRel (Han et al., 2018) and Wiki-ZSL (Chen and Li, 2021) demonstrate that ZETT effectively generalizes to the zero-shot setting, outperforming state-of-the-art methods by up to 6 points in accuracy. We find that ZETT shows lower variance in performance compared to existing methods that rely on fine-tuning on noisy synthetic data. We also show that it is robust to the choice of template and can be integrated with automatically generated templates without significant loss in performance. In conclusion, ZETT is an effective and efficient method for the zero-shot triple extraction task.

2 Method

We now introduce ZETT (**ZE**ro-shot **T**riplet extraction by **T**emplate infilling), a generative triplet extraction framework that reformulates triplet extraction as template infilling.

2.1 Task Definition

The goal of triplet extraction is to extract triplets $T = \{(h, r, t)\}$ given an input context c, where h and t are head and tail entities and r is a predefined relation $r \in \mathcal{R}$. In the zero-shot setting, we only have access to the dataset with the subset of relations $\mathcal{R}_{\text{seen}} \subset \mathcal{R}$ during training and have to generalize it to the dataset with unseen relations $\mathcal{R}_{\text{unseen}} \subset \mathcal{R}$ which is disjoint from seen relations: $\mathcal{R}_{\text{seen}} \cap \mathcal{R}_{\text{unseen}} = \phi$ (Chia et al., 2022).

2.2 Triplet Extraction by Template Infilling

We reformulate the triplet extraction task to a template infilling task as follows. For each relation $r \in \mathcal{R}$, we create a template τ_r with placeholders for the head (<head>) and tail (<tail>) entities, and then fill in these placeholders with entities from the context c to identify the head (h) and tail (t) entities for the relation r. For example, for the relation *participant in*, we prepare a template, "<head> is a participant in <tail>". Using a context c ("His brother Byron LaBeach, also a sprinter, competed in the 1952 Summer Olympics representing Jamaica."), we fill the placeholders in the template to identify the head ("Byron LaBeach") and tail ("1952 Summer Olympics") entities for the relation *participant in*. We show more template examples in Table 1.

With this formulation, we can even extract unseen relations $\mathcal{R}_{\text{unseen}}$ simply by preparing their templates and infilling them. This eliminates the need for additional fine-tuning for the unseen relations. At inference, we perform template infilling
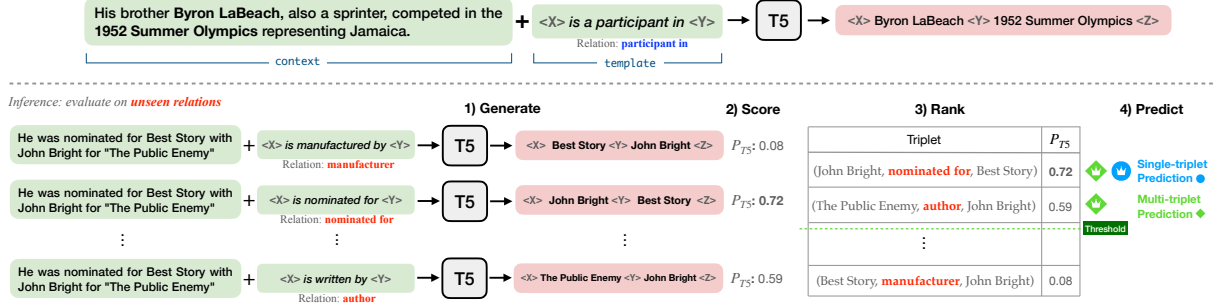
Figure 2: Fine-tuning and inference in ZETT: it fine-tunes T5 by extending input text with a relation template and learning to predict entity spans masked in the template. At inference, given templates for all unseen relation, it generates their entity spans and scores them. The best scoring sequences over a given threshold are then produced as the final output.

Relation	Description	Template
participant in	event in which a person or organization was/is a participant	`<head>` is a participant in `<tail>`.
publisher	organization or person responsible for publishing books, periodicals, printed music, podcasts, games or software	`<head>` is published by `<tail>`.
screenwriter	person(s) who wrote the script for subject item	`<tail>` wrote the script for `<head>`.
cast member	actor in the subject production	`<tail>` is an actor in `<head>`.

Table 1: Examples of relation types and templates. `<head>` and `<tail>` denote head and tail entities in a triplet. Each template is created based on its description in Wikidata (Vrandečić and Krötzsch, 2014). We provide the full list of templates in supplementary materials.

for all target unseen relations and re-rank them based on the consistency between the context c and the infilled template.

2.3 ZETT implementation with T5

We instantiate ZETT using the pre-trained language model T5 (Raffel et al., 2019). T5 is pre-trained to predict consecutive spans randomly dropped out from a sentence, which is closely aligned with the template infilling task.

As illustrated in Figure 2, we build input-output pairs for the text infilling task using the context c, the gold triplet $T = (h, r, t)$, and corresponding template τ_r. We replace the placeholder tokens `<head>` and `<tail>` in the template τ_r with mask tokens `<X>` and `<Y>`,[2] and concatenate it with the context c to form a model input x. Then, we fine-tune the model to learn to generate output y consisting of the gold head and tail entities where

each entity follows the corresponding mask tokens. We use the standard negative log-loss minimization $\mathcal{L} = -\log P_{T5}(y \mid x)$ for fine-tuning.

2.4 Inference with relation constraint

At inference, we evaluate the model on the test data wherein input contexts have unseen relations $\mathcal{R}_{unseen}$ during training. Given a context c, we build multiple model inputs $\{x_r\}_{r \in \mathcal{R}_{unseen}}$ by concatenating templates τ_r for each $r \in \mathcal{R}_{unseen}$. We then generate entity tokens for each sequence. Since some contexts may have multiple triplets, we use beam search to generate multiple output sequences for each model input x_r. We then compute a score for each output sequence as $P_{T5}(y \mid x_r)$. This score is used to rank all the generated triplets. The triplets are evaluated under single-triplet and multi-triplet settings. Under single-triplet setting, it is assumed that the input sentence has one triplet. In this setting, we predict the best scoring triplet as the output. Under multi-triplet setting, a sentence can have more than one triplet. In that case, we use threshold over the score to filter triplets. We tune the threshold on the validation set.

Exhaustively generating sequences for every unseen relation and scoring them can be inefficient in real-world scenarios where the number of un-

---

[2]In T5 implementation, mask tokens are denoted as `<extra_id_n>`, where $n \in \{0, ..., 99\}$. We use simplified forms `<X>`, `<Y>`, and `<Z>` instead of `<extra_id_0>`, `<extra_id_1>`, and `<extra_id_2>`. We also note that `<X>` and `<Y>` are not respectively corresponding to `<head>` and `<tail>`. `<X>` and `<Y>` are used as mask tokens in the input sequence and as delimiters for predicted spans in the output sequence. Thus `<X>` always comes first followed by `<Y>`. On the other hand, the order of `<head>` and `<tail>` depends on its template.

	# of examples	# of relations	# of entities
FewRel	56,000	80	72,954
Wiki-ZSL	94,383	113	77,623

$|\mathcal{R}_{\text{train}}|$		$|\mathcal{R}_{\text{test}}|$	$|\mathcal{R}_{\text{validation}}|$
FewRel	Wiki-ZSL		m
70	103	5	5
65	98	10	5
60	93	15	5

Table 2: Statistics of the datasets. $|\mathcal{R}|$ denotes the number of relations in each set, and m is the number of relations in the test set.

seen relations is large. Intuitively, not all unseen relations are related to the input context. Moreover, since ZETT relies on the LM's probability P_{T5}, the model tends to assign a higher score when the relation is frequently used in common sentences even when it is not relevant to the context. Therefore, instead of exhaustive scoring, we exploit relation constraints to filter out relations which are irrelevant to the given context. Since we don't have any data for unseen relations, we adopt the relation extraction model that utilizes relation descriptions (Chen and Li, 2021). We use a sentence similarity score between the context c and the description about the relation r to exclude irrelevant relations from $\mathcal{R}_{\text{unseen}}$. For the relation descriptions, we use the descriptions from Wikidata as shown in Table 1. Before generating entities, we first obtain the sentence embedding of the context and the relation's description using the off-the-shelf SBERT (Reimers and Gurevych, 2019). Then, we compute the cosine similarities between the context and relation's description embeddings and set the threshold δ on the validation set to filter out the relations whose similarity score is lower than δ. After filtering out the irrelevant relations, we evaluate the model on the constrained unseen relation set.

3 Experiments

3.1 Dataset

We evaluate our method on two datasets: FewRel (Han et al., 2018) and Wiki-ZSL (Chen and Li, 2021). FewRel is designed primarily for few-shot relation extraction. Wiki-ZSL is a subset of Wiki-KB and targets zero-shot relation extraction. Both datasets are created using distant supervision, but FewRel has additionally been filtered by humans. We use dataset versions released by Chia et al. (2022) which have been transformed for zero-shot

triplet extraction. We follow their zero-shot set up for training and evaluation as follows: 1) we keep relation types in training, validation, and test splits disjoint, 2) we evaluate different methods under different settings for the size of unseen relation types ($m \in \{5, 10, 15\}$), 3) To avoid experimental noise, we repeat experiments using different data folds wherein relation types are split with different random seeds: $\{0, 1, 2, 3, 4\}$. Table 2 shows statistics of each dataset and setting.

3.2 Experimental Settings

Training We use the pre-trained T5-base[3]. We fine-tune the model for 3 epochs with 64 batch size, and 3e-5 learning rate. We tune the parameters using the validation set. We provide details of the parameters in Appendix.

Inference At inference, ZETT generates entity spans given the input sentence concatenated with the relation template. We restrict the vocabulary to use tokens from the input sentence to ensure entities are spans from the input sentence. We use a beam size of 4 to generate a maximum of 4 candidate entity pairs for each relation type. For the relation constraint, we set a threshold δ and choose δ=0.85 based on the validation performance.

Single- and Multi-triplet evaluation Each example in the datasets includes one or more triplets. We evaluate the models separately on single- and multi-triplet settings following the previous study (Chia et al., 2022). For the single-triplet setting, the examples have only one correct (gold) triplet, thus we use accuracy as the metric for evaluating performance. In the multi-triplet setting, the number of gold triplets is two or more, thus we evaluate performance with a F1 score. To retrieve positives in the multi-triplet setting, we set a threshold and output a candidate as a predicted positive example if its score is above this threshold. The thresholds for the multi-triplet evaluation are provided in Appendix.

3.3 Baseline methods

We compare the performance of ZETT with four existing methods for triplet extraction: **1) ZS-BERT+spanNER** is a pipeline model, where two sub-tasks: relation classification and entity extraction are combined to extract triplets. We use ZS-BERT (Chen and Li, 2021) for the relation classi-

---
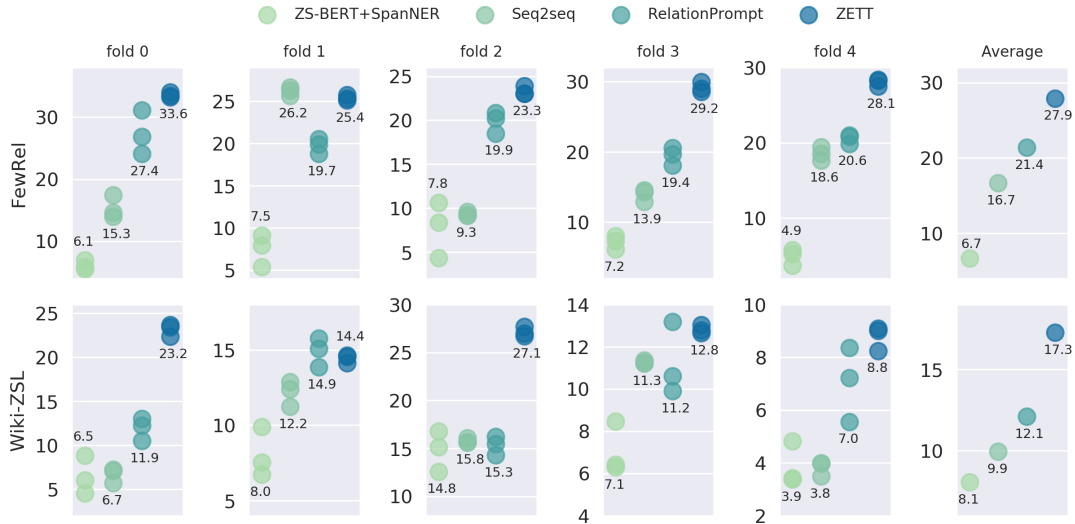
[3] https://huggingface.co/t5-base

Figure 3: Results of each data fold in the single-triplet setting with m=10. We experiment on three different random seeds. Each point is the accuracy (%) of each evaluation and labeled numbers are an average of three results. Final column shows average over all folds.

| | Model | Single-triplet (Accuracy (%)) | | | Multi-triplet (F1 score (%)) | | |
		m=5	m=10	m=15	m=5	m=10	m=15
FewRel	TableSequence[†]	11.82	12.54	11.65	3.40	6.37	3.48
	ZS-BERT+spanNER	7.22 (±0.67)	6.68 (±1.48)	7.68 (±0.23)	15.10 (±2.52)	12.14 (±2.33)	14.78 (±1.81)
	Seq2Seq	22.02 (±1.07)	16.71 (±0.82)	11.91 (±0.70)	26.02 (±2.81)	17.12 (±1.86)	13.02 (±0.96)
	RelationPrompt	24.36 (±0.90)	21.45 (±1.50)	20.24 (±0.72)	30.85 (±3.23)	24.45 (±1.76)	23.65 (±1.36)
	ZETT	**30.71** (±0.45)	**27.90** (±0.31)	**26.17** (±0.20)	**33.71** (±0.42)	**31.28** (±0.54)	**24.39** (±0.37)
Wiki-ZSL	TableSequence[†]	14.47	9.61	9.20	6.29	6.4	6.39
	ZS-BERT+spanNER	8.16 (±1.26)	8.05 (±0.79)	6.47 (±0.42)	15.96 (±1.11)	11.98 (±2.28)	11.90 (±0.84)
	Seq2Seq	14.73 (±1.30)	9.94 (±0.46)	7.05 (±0.44)	30.71 (±4.31)	19.70 (±1.90)	11.52 (±3.32)
	RelationPrompt	16.74 (±1.53)	12.13 (±0.86)	10.47 (±0.96)	**33.28** (±1.70)	24.04 (±2.12)	18.73 (±1.98)
	ZETT	**21.49** (±0.44)	**17.27** (±0.31)	**12.78** (±0.42)	31.17 (±0.69)	**24.87** (±0.32)	**21.21** (±0.35)

Table 3: Accuracy in the single-triplet and F1 score in the multi-triplet settings. m denotes the number of unseen relations. All reported results are averaged over three different random seeds, where the result of each seed are averaged over five different data folds. Results of † are taken from Chia et al. (2022).

fication model and implement span-based named entity recognition model using transformer encoder initialized with `roberta-base` checkpoint for entity extraction (Fu et al., 2021). **2) TableSequence** (Wang and Lu, 2020) is a joint learning model with two separate encoders performing relation extraction and named entity recognition at the same time. Since TableSequence is designed for supervised learning, we report the results of models trained on synthetic data from Chia et al. (2022). **3) Seq2seq** (Chia et al., 2022) is an encoder-decoder model based on the pre-trained BART-base (Lewis et al., 2020). The input to the encoder is a context, then the decoder generates a triplet as a sequence of structured template. **4) RelationPrompt** (Chia et al., 2022) is an additionally fine-tuned model of the Seq2Seq on the synthetic data for the un-

seen relations. [4] They generate synthetic training datasets for unseen relations using GPT-2 (Radford et al., 2019)[5] and fine-tune the Seq2seq-based triplet extraction model.

4 Results

4.1 Automatic Evaluation

Table 3 shows the results of single- and multi-triplet evaluation settings on FewRel and Wiki-ZSL. As can be seen, ZETT consistently outperforms existing methods across different datasets under single-

---

[4]Since the synthetic data and pre-trained checkpoints for all experiments are not provided in the official implementation, we couldn't reproduce the results of RelationPrompt as same as reported in the paper. Thus, we re-trained the models with three different random seeds and report the average of them in Table 3.

[5]https://huggingface.co/gpt2

Model	Single-triplet (Accuracy (Δ))		Multi-triplet (F1 score (Δ))	
	FewRel	Wiki-ZSL	FewRel	Wiki-ZSL
RelationPrompt	21.45	14.37	24.45	24.04
ZETT w/ *Manual*	27.90	17.27	31.28	24.87
ZETT w/ *Paraphrased* (Random)	24.74 (-3.16)	14.49 (-2.78)	28.86 (-2.42)	23.39 (-1.48)
ZETT w/ *Paraphrased* (Top 1)	25.12 (-2.78)	15.34 (-1.93)	29.60 (-1.68)	24.63 (-0.24)

Table 4: Comparison of performance with paraphrased templates at m=10. Δ is the performance difference over ZETT with manual templates.

triplet setting. It achieves up to 6.45 and 5.14 higher accuracy than the existing state-of-the-art model, RelationPrompt on FewRel and Wiki-ZSL datasets, respectively. It also shows much lower variance in performance than RelationPrompt (see Figure 3). In the worst case, performance of RelationPrompt differs by 7.1 points in accuracy (fold 0, m=10, FewRel). We conjecture that the variance can be attributed to the varying quality of the synthesized dataset in every trial. On the the other hand, ZETT shows stable performance through all trials, consistently outperforming the existing methods. In the multi-triplet setting, ZETT achieves the best F1 score for different relation set sizes except on Wiki-ZSL with m=5. We argue that this is mainly due to the biased distribution in the multi-triplet test sets; most examples are only for few relations. Therefore, performance loss in a particular relation results in significant drop in overall performance. We analyze the main causes of prediction failure in Section 5.4. Overall, results of automatic evaluation show that having a simpler training process can yield more effective and stable performance on the task.

4.2 Human Evaluation

Both the test sets from our experiments are created using distant supervision and hence can be noisy and incomplete. In other words, they can include triplets that are not supported by an input text. Furthermore, it may not include all possible triplets from an input text. We, therefore, conduct human evaluation to better understand the performance of ZETT. We focus on WikiZSL since it is noisier and sample 200 contexts for manual annotation. We ask three CS graduates to annotate top-5 predictions for each context. An annotator labels a prediction correct if it is supported by the input text. Three annotators labeled the triplets such that each triplet receives two annotations. We identified triplets labeled as *True* by both annotators, showing a high agreement with a Cohen's Kappa coefficient

of 0.75. Among the 1,000 triplets, we found 127 mislabeled instances–24 were *False* (i.e., unsupported by input text) but originally labeled as *True*, and 103 were *True* but not covered in the original dataset. We found that accuracy of ZETT increased from 18% to 30.2% on this manually annotated dataset. We will release the manually annotated dataset for future research.

5 Discussion

5.1 Robustness to choice of templates

Since ZETT uses templates to generalize to unseen relations, the performance can be sensitive to the wording of templates (Sanh et al., 2022). In this section, we investigate how the performance varies depending on the template. To that end, we paraphrase the manually defined templates and compare the performance of ZETT with manual and paraphrased templates.

We adopt the back-translation method from Jiang et al. (2020), which uses an English-German machine translation model to generate semantically-similar templates. In our experiments, we used the translation model to generate 7 templates in German language for each manual template, and then back-translate each of them to 7 templates in English. As a result, we obtain 49 paraphrased templates per relation. Since many of them are duplicates, we evaluate on the most-frequent template, *Paraphrased* (Top 1). We also compare with a randomly selected paraphrased template, *Paraphrased* (Random). We will release the full set of paraphrased templates.

Table 4 compares the performance of ZETT with manual and paraphrased templates. Since automatically paraphrased templates can be noisy in capturing the semantic meaning of a relation, we observe small performance drops when using ZETT with paraphrased templates. The performance drop is much smaller with *Paraphrased* (Top 1). Even though the paraphrased templates are noisy, ZETT

	CONTEXT: Jimmy Jam is the son of Cornbread Harris, a Minneapolis blues and jazz musician.		
	GOLD TRIPLET: (Jimmy Jam, *father*, Cornbread Harris)		
RP	MODEL OUTPUT	Head Entity : Cornbread Harris , Tail Entity : Minneapolis blues, Relation : field of work .	
	TRIPLET	(Cornbread Harris, field of work, Minneapolis blues)	
ZETT	TEMPLATE	`<tail>` is a father of `<head>`	
	MODEL OUTPUT	`<X>` Cornbread Harris `<Y>` Jimmy Jam `<Y>`	
	TRIPLET	(Jimmy Jam, *father*, Cornbread Harris)	

	CONTEXT: He participated in UEFA Euro 1972 for the Hungary national football team.		
	GOLD TRIPLET: (UEFA Euro 1972, *participating team*, Hungary national football team)		
RP	MODEL OUTPUT	Head Entity : Hungary national football team , Tail Entity : UEFA Euro 1972, Relation : participating team .	
	TRIPLET	(Hungary national football team, *participating team*, UEFA Euro 1972)	
ZETT	TEMPLATE	`<tail>` is a participating team in `<head>`	
	MODEL OUTPUT	`<X>` Hungary national football team `<Y>` UEFA Euro 1972 `<Y>`	
	TRIPLET	(UEFA Euro 1972, *participating team*, Hungary national football team)	

Table 5: Example MODEL OUTPUT sequences and triplets from RelationPrompt (RP) and ZETT.

still consistently outperforms current state-of-the-art methods.

5.2 Ablation Study

Next, we conduct an ablation study to examine the importance of each generation setting and the relation constraint. The results are summarized in Table 6. First, we test without the vocabulary constraints that limits the vocabulary set to tokens that appear in the context. We observe a small drop in accuracy of up to 0.39 points. Second, we compare the performance of beam search with greedy decoding that only generates one sequence. We find that beam search improves accuracy by up to 1.62 points since it selects the best complete sequence of entity pair as opposed to selecting the best individual entity token in each position. Last, we observe that our proposed relation constraint, although simple, is effective in eliminating irrelevant relations and can improve accuracy by up to 3.56 points.

5.3 Qualitative Analysis

To further gain insights into the strengths and weaknesses of the models, we manually inspect examples where ZETT and RelationPrompt generate different triplets. Table 5 shows some such examples. We observe that RelationPrompt often generates an incorrect triplet even for easy examples. For example, it fails to predict the correct relation *father* even when a context ("Jimmy Jam is the son of Cornbread Harris") clearly describes it. This can be attributed to noise in the synthesized training data for unseen relations. We find the training data for relation *father* includes noisy examples

such as "The Last Days of the Lion-Cat is based on David Simon." → (The Last Days of the Lion-Cat, *father*, David Simon), where the context has no information about *father*. Such noisy examples can propagate errors in the multi-step training process of RelationPrompt. ZETT, being a single-step process, is more robust to such errors.

We also find that RelationPrompt often fails to predict the correct order of entities. This is because the output sequence for generating a triplet "Head Entity: `<head>`, Tail Entity : `<tail>`, Relation : `<relation>`" does not encode any information about the order of entities. In contrast, ZETT can leverage the implicit information about entity types and their order encoded in relation templates. For instance, for the relation *participating team*, the template "`<tail>` is a participating team in `<head>`" provides implicit information that `<head>` entity is a sports team and `<tail>` entity should be a contest or sports game. This information helps ZETT correctly predict the order of entities.

5.4 Error Analysis

To understand the limitations of ZETT, we perform a detailed analysis of errors on 200 examples. Incorrect ranking of relations contributed to the most frequent errors, accounting for 36% of errors. Since ZETT relies on LM's probability P_{T5}, it tends to assign high scores to relations that are frequently used in common sentences. We find that relations such as *occupation*, *owned by* or *work location* had higher scores than more rare relations such as *contains administrative territorial entity* or *place served by transport hub*. Lack of discriminatory power over semantically similar relations

	FewRel	Wiki-ZSL
ZETT	27.90	17.27
w/o decoding vocab constraint	27.76	16.88
w/o beam search	26.87	15.65
w/o relation constraint	24.34	16.61

Table 6: Ablation study on m=10 and single-triplet setting. The metric is accuracy.

contributed to 21% of the errors. For example, relations such as *headquarters location*, *location of formation*, and *work location* all represent the concept of location and are hard for the model to discriminate without any fine-tuning. Lastly, we find that relation constraint sometimes excluded the correct relation, contributing to 17% of the errors. For instance, when the context "Ay was the penultimate Pharaoh of Ancient Egypt's 18th dynasty." and the gold triplet (Ay, *country of citizenship*, Ancient Egypt) are given, even though the context contains the information of Ay's citizenship, our model predict as (Ay, *occupation*, Pharaoh) by determining *country of citizenship* is irrelevant. The rest of error types include failure of predicting entity spans, flipped entity pair, generation of null string, and labeling errors of datasets. We provide more detailed examples of error types in Appendix. Future work can focus on developing more effective score functions and relation classification techniques.

6 Related Work

Zero- and few-shot Triplet Extraction   Triplet extraction has largely been studied as a pipeline of two sub-tasks: entity extraction and relation extraction, with pipeline (Zhong and Chen, 2021), joint-learning (Roth and Yih, 2004; Yu and Lam, 2010; Singh et al., 2013; Miwa and Sasaki, 2014; Li and Ji, 2014; Paolini et al., 2021) and end-to-end neural approaches (Zheng et al., 2017). Although open information extraction (Etzioni et al., 2008) shares a similar objective, it extracts relation spans from the input text which later have to be canonicalized to obtain relation types. Triplet extraction instead targets a predefined set of relation types. Zero-shot triplet extraction aims to generalize the models to an unseen set of canonicalized relation types. The task was proposed by Chia et al. (2022) that generalizes by learning to create synthetic data for unseen relations. However, their approach does not provide guarantees of quality and consistency of the synthetic data, making it hard to reproduce. Our task formulation that relies on generative

models is inspired by recent progress in relation extraction. Yang et al. (2021) shows that leveraging additional information about entities can yield better zero-shot and few-shot performance on relation extraction. Wang et al. (2021) shows that a unified framework based on text-to-triple model can achieve good zero-shot performance for open information extraction and relation classification tasks. Inspired by these observations, we propose a text-to-text approach for the triplet extraction task that leverages additional information encoded in relational templates to achieve state-of-the-art performance in zero-shot settings.

LM prompt-tuning with templates   Recent progress in LM prompt-tuning aims to bridge the gap between pre-training and downstream tasks by using natural language templates. Most approaches reframe the downstream task as a masked language modeling problem and have been successfully applied for text classification (Obamuyide and Vlachos, 2018; Hu et al., 2022; Chen et al., 2022), named entity recognition (Cui et al., 2021), and natural language inference (Schick and Schütze, 2021a,b). However, these approaches are mostly tailored for classification tasks, and not suitable for structured prediction such as triplet extraction. Hsu et al. (2022) introduced DEGREE, a method akin to ZETT, employing a template infilling task to extract events in a low-resource setting. However, DEGREE uses placeholders like "someone" or "somewhere" that are not part of the pre-training process, necessitating fine-tuning the model to new types of event. In contrast, the fine-tuning of ZETT is identical to pre-training of T5, that improves its capability to generalizability to unseen relations.

7 Conclusion

We introduced the ZETT, a new framework for zero-shot triplet extraction which does not need any data augmentation or pipeline systems. We reformulate the triplet extraction as a template infilling problem using natural language templates. This enables the model to better leverage PLMs by aligning pre-training, fine-tuning, and inference objectives and eliminates the need for additional training data for unseen relations. ZETT is effective in extracting triplet by leveraging knowledge in PLMs, and is also more stable with a simple training process. Through experiments on two datasets, we demonstrated that ZETT outperforms previous state-of-the-art methods, showing consistent perfor-

mance improvement without any extra models or synthetic data. We also showed that ZETT is robust in the variation of templates, showing competitive results without a significant performance loss.

Limitations

Method limitations   As discussed in Section 5.4, ZETT has several weaknesses: 1) the ranking score based on the PLM's probability is likely high when the templates of the relation include phrases or sentences which are commonly used in corpus. 2) The model struggles to discriminate between similar relations. 3) The relation constraint method proposed in Section 2.4 could exclude the relevant relations in inference. Future work could explore more effective relation classification methods for the relation constraint and sophisticated score functions which are well generalized to relations even whose templates are infrequent in corpus.

Zero-shot setup limitations   The zero-shot setup that has been explored in the literature assumes that the set of unseen relations is given in inference, and the number of unseen relations is at most 15. This setup cannot fully reflect the real-world problems where there are numerous unseen relations. Future work could include exploring more realistic task setups and developing techniques to overcome the challenges posed by them.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730, Dublin, Ireland. Association for Computational Linguistics.

Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. RelationPrompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 45–57, Dublin, Ireland. Association for Computational Linguistics.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online. Association for Computational Linguistics.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Jinlan Fu, Xuanjing Huang, and Pengfei Liu. 2021. SpanNER: Named entity re-/recognition as span prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7183–7195, Online. Association for Computational Linguistics.

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

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. DEGREE: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1890–1908, Seattle, United States. Association for Computational Linguistics.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2225–2240, Dublin, Ireland. Association for Computational Linguistics.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2017. Distant supervision for relation extraction

with sentence-level attention and entity descriptions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2181–2187. AAAI Press.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, Doha, Qatar. Association for Computational Linguistics.

Abiola Obamuyide and Andreas Vlachos. 2018. Zero-shot relation classification as textual entailment. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 72–78, Brussels, Belgium. Association for Computational Linguistics.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies, pages 2339–2352, Online. Association for Computational Linguistics.

Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. 2013. Joint inference of entities, relations, and coreference. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, AKBC '13, page 1–6, New York, NY, USA. Association for Computing Machinery.

Daniil Sorokin and Iryna Gurevych. 2017. Context-aware representations for knowledge base relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1784–1789, Copenhagen, Denmark. Association for Computational Linguistics.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2021. Zero-shot information extraction as a unified text-to-triple translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1225–1238, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on Freebase via relation extraction and textual evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2326–2336, Berlin, Germany. Association for Computational Linguistics.

Shan Yang, Yongfei Zhang, Guanglin Niu, Qinghua Zhao, and Shiliang Pu. 2021. Entity concept-enhanced few-shot relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 987–991, Online. Association for Computational Linguistics.

Xiaofeng Yu and Wai Lam. 2010. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In *Coling 2010: Posters*, pages 1399–1407, Beijing, China. Coling 2010 Organizing Committee.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods*

in Natural Language Processing, pages 1753–1762, Lisbon, Portugal. Association for Computational Linguistics.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2856–2878, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

A Appendix

A.1 Experimental Setup Details

batch size	64
learning rate	3e-5
warm-up ratio	0.2
maximum input length	128 {128, 256}
maximum output length	64 {64, 128, 256}
beam size	4 {1, 2, 4, 8, 16}

Table 7: Best-performing hyperparameters and search space. Values in parentheses denote search space.

	Threshold	
m	FewRel	WikiZSL
5	-2.6	-2.6
10	-2.5	-2.5
15	-2.5	-2.6

Table 8: Best-performing thresholds in the multi-triplet evaluation. The triplets with scores ($\log P_{T5}$) above the threshold are considered final predictions.

Model	# params
ZS-BERT+spanNER	224M
TableSequence (Wang and Lu, 2020)	240M
Seq2seq (Chia et al., 2022)	140M
RelationPrompt (Chia et al., 2022)	140M
ZETT	220M

Table 9: The number of parameters in each model.

Hyperparameter Table 7 describes hyperparameters and search spaces we considered in experiments. In training, we used AdamW optimizer (Loshchilov and Hutter, 2019) for all transformer-based models with 0.2 warm-up ratio.

Thresholds of the multi-triplet evaluation As mentioned in Section 2.4, we use a threshold to retrieve positives when the example includes two or more triplets. We repeated the evaluation with the threshold in range of {2.0, 2.1, 2.2, . . . , 3.4, 3.5} on the validation set and chose the best performing ones. We report the detailed values in Table 8.

Computing Infrastructure We ran all experiments on a single NVIDIA GeForce GTX 1080 (12GB) with CUDA 10.1 version.

Computational Budget Training ZETT with hyperparameters in Table 7 takes 1.5h on the FewRel dataset and 2h on the Wiki-ZSL dataset with a single NVIDIA GeForce GTX 1080.

Model Parameters Table 9 provides the number of parameters in each model used in our experiments.

A.2 Guidelines for Human Evaluation

The goal of the human evaluation is to address that the datasets have some wrong labeled examples and false negatives. We manually evaluated 1,000 examples: Top 5 predictions (triplets) of ZETT from 200 contexts. The contexts are randomly sampled from the single-triplet test set of m=10 in Wiki-ZSL. Three annotators are given 667, 667, and 666 examples, respectively. Each example consists of input sequence (context and template), gold triplet, predicted triplet, and the model's score P_{T5}. The instructions for annotation are as follows:

- 1) Annotate **TRUE** if you think the given context and the model's prediction (triplet) are matched. Otherwise annotate **FALSE**.

- 2) Annotate **FALSE** if we cannot infer the triplet from the given context, even if the triplet itself is true. Concretely, for the example of the context "Elected to the comptrollers post in 1998 as a Republican , Keeton ran as an independent candidate for Texas governor against Republican incumbent James Richard Rick Perry in 2006." and the triplet (James Richard Rick Perry, *residence*, Texas), We cannot be sure whether James Richard Rick Perry lives in Texas or not just given the context. Thus, we should label this as **FALSE**.

We also provided the relation descriptions to avoid confusion between similar relations. For example, the relation *residence* and *location* can be confusing to annotators since both relations can refer to places of something. However, when we refer the description, we can clarify that these two relations have different head entity types: person for *residence*, and {object, structure or event} for *location*.

- *residence* (P551): the place where the person is or has been, resident

- *location* (P276): location of the object, structure or event. In the case of an administrative entity as containing item use P131. For statistical entities use P8138. In the case of a geographic entity use P706. Use P7153 for locations associated with the object.

a) CONTEXT: This lizard lives in the southwestern part of Africa, in Namibia and South Africa.	
LABELED TRIPLET: (South Africa, *shares border with*, Namibia)	
b) CONTEXT: He studied photography at the International Center of Photography in New York City in 1990, under the tutelage of Larry Clark and Nan Goldin.	
LABELED TRIPLET: (International Center of Photography, *headquarters location*, New York City)	
c) CONTEXT: FC Slavutych was a Ukrainian football club from Slavutych, Kiev Oblast.	
LABELED TRIPLET: (Ukrainian, *contains administrative territorial entity*, Kiev Oblast)	
d) CONTEXT: Google released Android 7.1.1 Nougat for the Pixel C in December 2016.	
LABELED TRIPLET: (Pixel C, *operating system*, Android)	

Table 10: Wrong labeled examples in the FewRel dataset. In example a), b), and c), although these triplets are true, the contexts are not related to the given relations or the triplets cannot be inferred from the contexts. Another type of error is a false negative. In the example d), our model predicts (Pixel C, *owned by*, Google) given the context, and this knowledge is also true. However, the triplet (Pixel C, *owned by*, Google) is not labeled as an answer in the dataset.

Error type: Incorrect ranking

Context: Mondoedo is a small town and municipality in the Galician province of Lugo, Spain. Gold Triplet: (Galician, *contains administrative territorial entity*, province of Lugo)

Input		Prediction (head, *relation*, tail)	$\log P_{T5}$
Context	Template		
Mondoedo is a small town and municipality in the Galician province of Lugo, Spain.	<X>'s job is <Y>	(Mondoedo, *occupation*, municipality)	-1.05
Mondoedo is a small town and municipality in the Galician province of Lugo, Spain.	<X> contains administrative territorial entity <Y>	(Galician, *contains administrative territorial entity*, province of Lugo)	-1.52

Context: Its hub is Tinson Pen Aerodrome in Kingston (KTP), and its other major gateway was Sangster International Airport in Montego Bay (MBJ). Gold Triplet: (Tinson Pen Aerodrome, *place served by transport hub*, Kingston)

Input		Prediction (head, *relation*, tail)	$\log P_{T5}$
Context	Template		
Its hub is Tinson Pen Aerodrome in Kingston (KTP), and its other major gateway was Sangster International Airport in Montego Bay (MBJ).	<X> worked in <Y>	(Tinson Pen Aerodrome, *work location*, Kingston)	-0.78
Its hub is Tinson Pen Aerodrome in Kingston (KTP), and its other major gateway was Sangster International Airport in Montego Bay (MBJ).	<X> is headquartered in <Y>	(Sangster International Airport, *headquarters location*, Montego Bay)	-0.99
Its hub is Tinson Pen Aerodrome in Kingston (KTP), and its other major gateway was Sangster International Airport in Montego Bay (MBJ).	<X> is a place served by transport hub <Y>	(Tinson Pen Aerodrome, *place served by transport hub*, Kingston)	-1.71

Error type: Lack of ability to discern semantically similar relations

Context: Similar movements concurrently formed in many other countries, leading to the formation, at a 1947 meeting in Montreux, Switzerland, of a global coalition, now called World Federalist Movement.
Gold Triplet: (World Federalist Movement, *location of formation*, Montreux)

Input		Prediction (head, *relation*, tail)	$\log P_{T5}$
Context	Template		
Similar movements concurrently formed in many other countries, leading to the formation, at a 1947 meeting in Montreux, Switzerland, of a global coalition, now called World Federalist Movement.	<X> is headquartered in <Y>	(World Federalist Movement, *headquarters location*, Montreux)	-0.89
Similar movements concurrently formed in many other countries, leading to the formation, at a 1947 meeting in Montreux, Switzerland, of a global coalition, now called World Federalist Movement.	<X> was established in <Y>	(World Federalist Movement, *location of formation*, Montreux)	-1.74
Similar movements concurrently formed in many other countries, leading to the formation, at a 1947 meeting in Montreux, Switzerland, of a global coalition, now called World Federalist Movement.	<X> was established in <Y>	(World Federalist Movement, *location of formation*, Montreux, Switzerland)	-1.94
Similar movements concurrently formed in many other countries, leading to the formation, at a 1947 meeting in Montreux, Switzerland, of a global coalition, now called World Federalist Movement.	<X> worked in <Y>	(World Federalist Movement, *work location*, Switzerland)	-1.98

Figure 4: Examples according to error types. The first line of each table denotes the final prediction.