Graph-based Model Generation for Few-Shot Relation Extraction

Wanli Li and **Tieyun Qian***
School of Computer Science, Wuhan University, China
{wanli.li, qty}@whu.edu.cn

https://github.com/NLPWM-WHU/GM_GEN

(EMNLP-2022)

Reported by Ke Gan

1. Introduction
2. Approach
3. Experiments

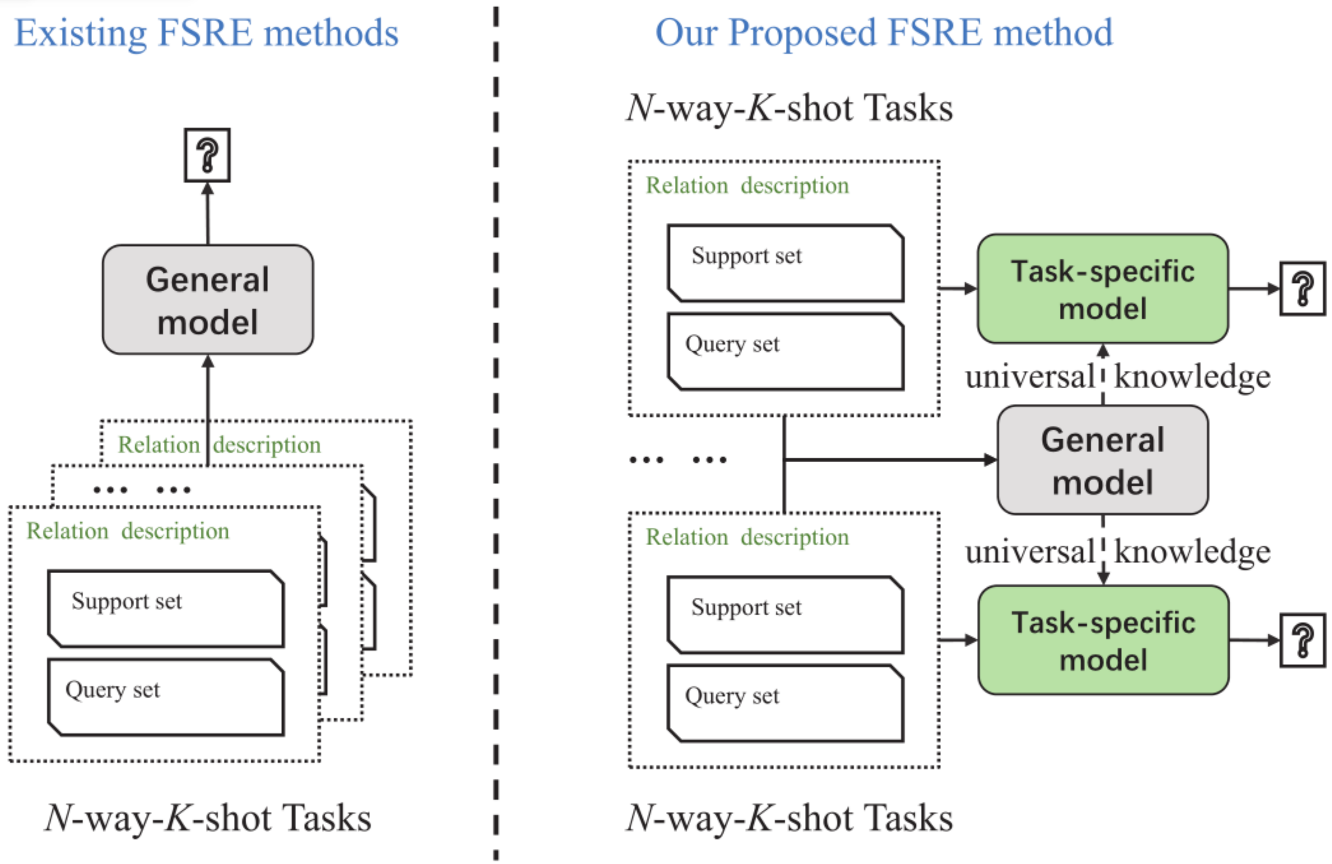
Introduction

Supporting Set	
(A) capital_of	(1) *London* is the capital of *the U.K.* (2) *Washington* is the capital of *the U.S.A.*
(B) member_of	(1) *Newton* served as the president of *the Royal Society*. (2) *Leibniz* was a member of *the Prussian Academy of Sciences*.
(C) birth_name	(1) *Samuel Langhorne Clemens*, better known by his pen name *Mark Twain*, was an American writer. (2) *Alexei Maximovich Peshkov*, primarily known as *Maxim Gorky*, was a Russian and Soviet writer.
Test Instance	
(A) or (B) or (C)	*Euler* was elected a foreign member of *the Royal Swedish Academy of Sciences*.

Given a piece of text $d = (w_1, w_2, ..., w_n)$, a subject entity $\tilde{e}_s$ and an object entity $\tilde{e}_o$

the task of RE is to predict the relation $r \in \mathcal{R}$ between $\tilde{e}_s$ and $\tilde{e}_o$, where $R = \{r_1, ..., r_{|R|}\}$ is a predefined relation set.

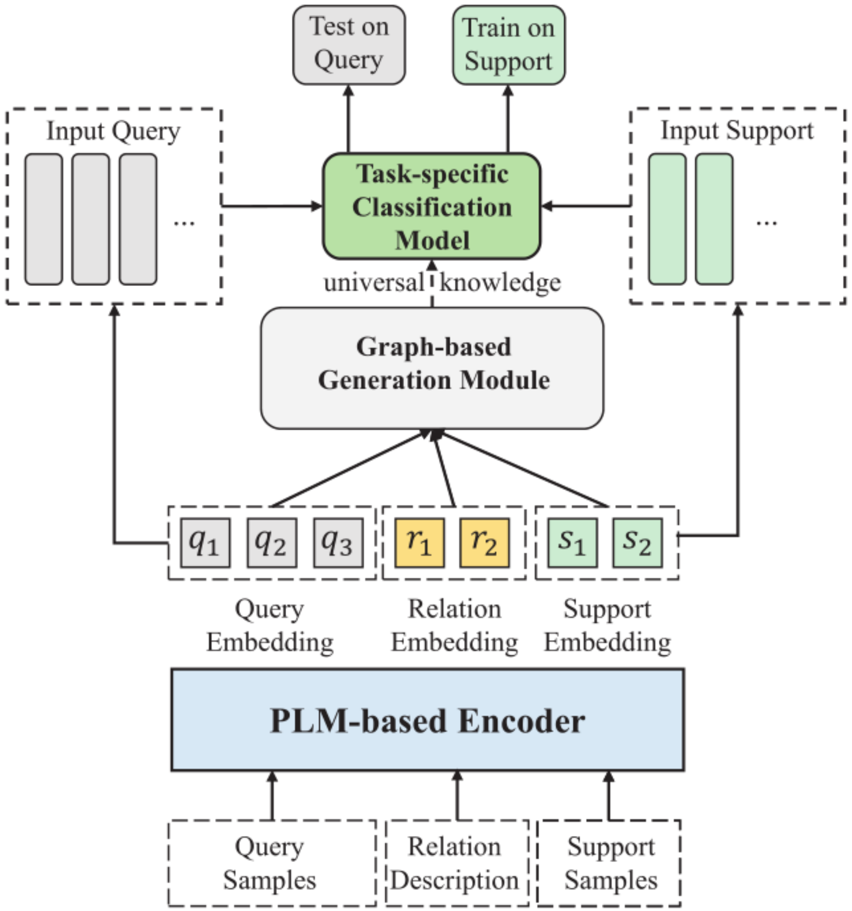
Introduction



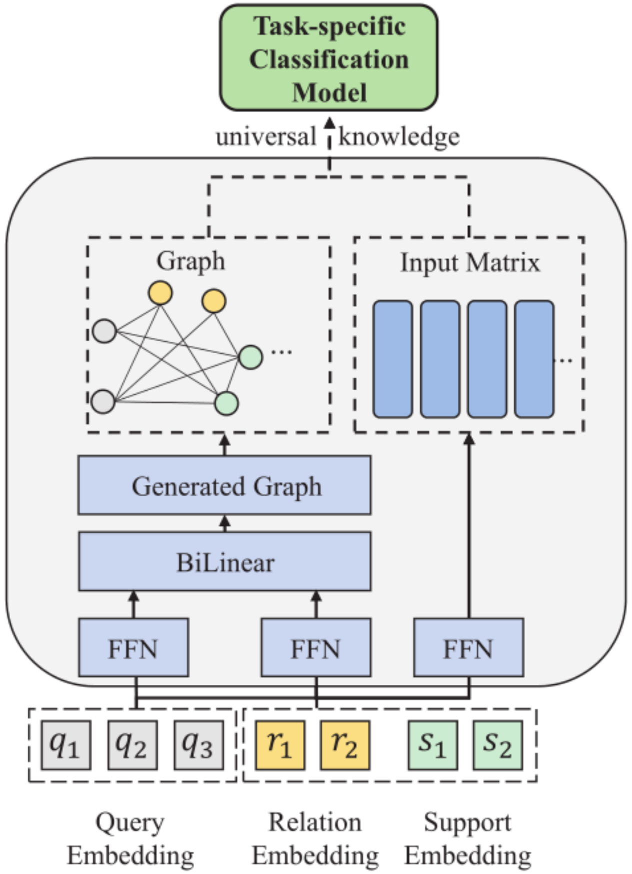
one general model for all tasks

decouple the complexity of the whole task space from that of each task space

Approach

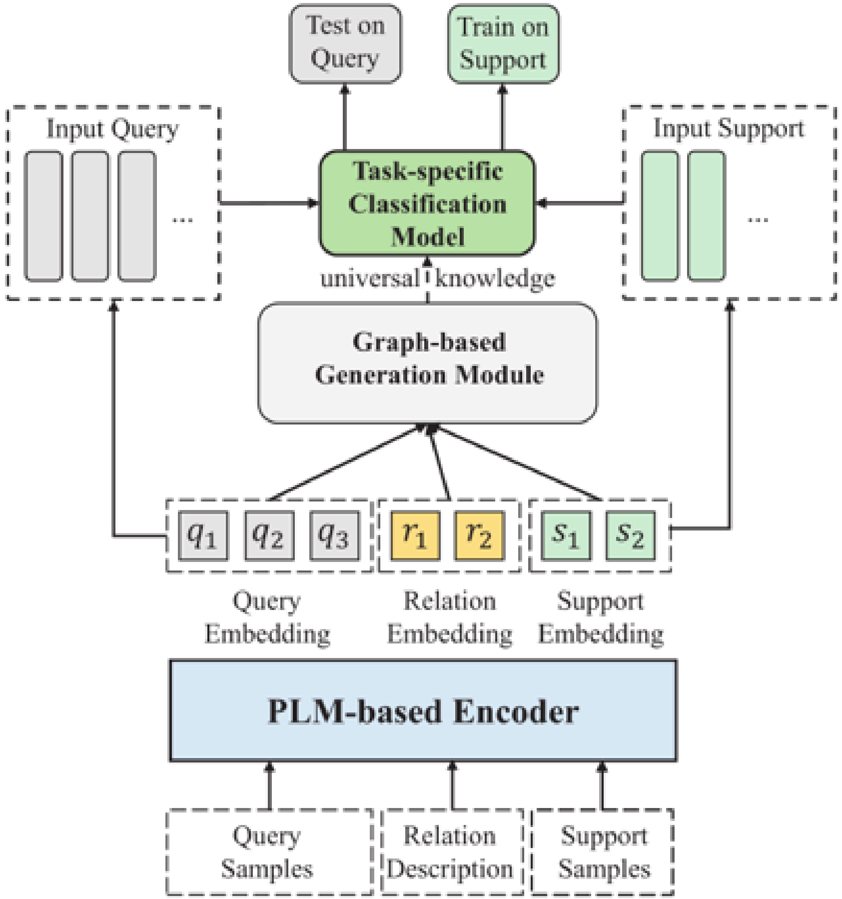


(a) Model Overview



(b) Graph Model Based Generation Module

Approach



(a) Model Overview

$$d = (w_1, w_2, ..., w_n)$$

insert the '[CLS]' and '[SEP]'

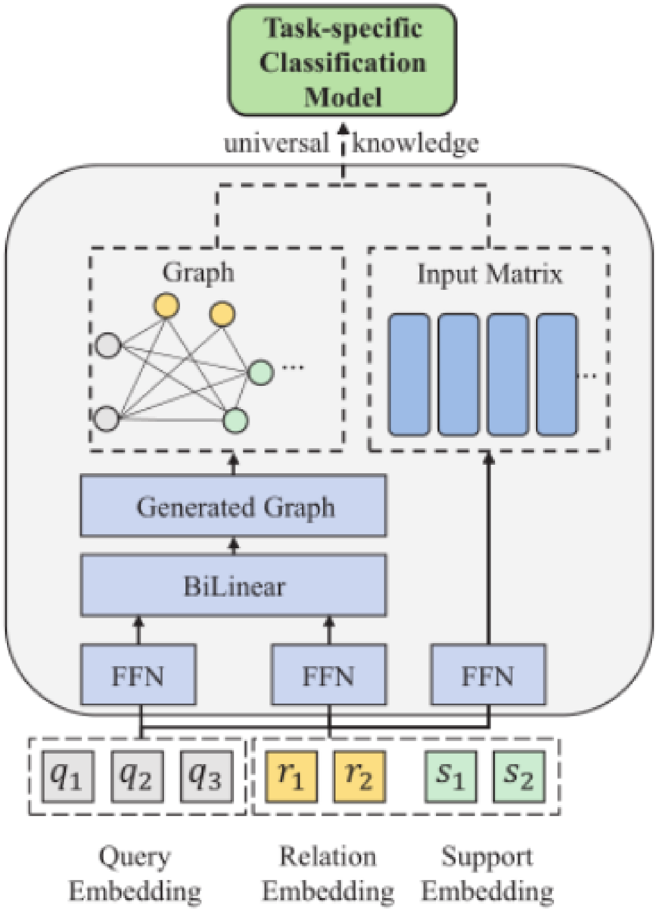
add four special tokens '[E1][/E1]' and '[E2][/E2]' at the beginning and end of the head entity and tail entity

$$\boldsymbol{x} = \boldsymbol{h}_i \oplus \boldsymbol{h}_j, \qquad (1)$$

$$\boldsymbol{x_{rel}} = \boldsymbol{h}_0 \oplus \frac{\sum_{j=1}^{N} \boldsymbol{h}_i}{N}, \qquad (2)$$

where h_0 represents the output of the '[cls]'

Approach



(b) Graph Model Based Generation Module

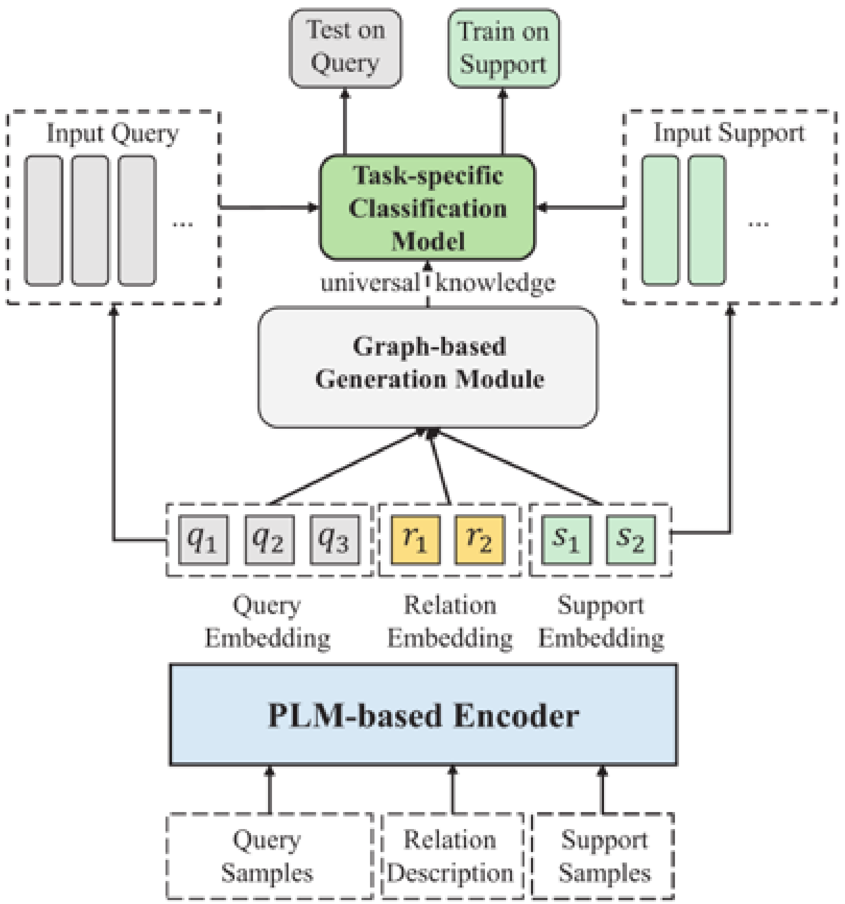
$$a_{ij} = \begin{cases} \text{bilinear}(\text{FFN}(\boldsymbol{x}_i), \text{FFN}(\boldsymbol{x}_j)), & i \neq j \\ 0, & i = j \end{cases}, \quad (3)$$

$$\boldsymbol{z}_i = \sigma(\sum_{j=1}^{n} \boldsymbol{A}_{ij} \boldsymbol{W} \boldsymbol{x}_j + b), \quad (4)$$

$$\boldsymbol{o_r} = \boldsymbol{z}_r \oplus \frac{\sum_{j \in S_r} \boldsymbol{z}_j^s}{n}, \quad (5)$$

The final output matrix $\boldsymbol{O} = \{o_1, ..., o_{|r|}\}$ contains general knowledge for inferring subsequent tasks and will be treated as the weight matrix to pass the universal knowledge

Approach



(a) Model Overview

$$\boldsymbol{C}^{\boldsymbol{q}} = \boldsymbol{x}^{q} \oplus \mathrm{FFN}(\boldsymbol{x}^{q}). \tag{6}$$

$$\tilde{\boldsymbol{y}}_q = \sigma(\boldsymbol{O}\boldsymbol{C}^{\boldsymbol{q}} + b). \tag{7}$$

$$\mathcal{L} = -\sum_{q \in D_{base}} \boldsymbol{y}_q \cdot \log(\tilde{\boldsymbol{y}}_q), \tag{8}$$

Experiment

Encoder	Model	5-1	5-5	10-1	10-5
CNN	Proto-CNN (Snell et al., 2017b)	72.65/74.52	86.15/88.40	60.13/62.38	76.20/80.45
	Proto-HATT (Gao et al., 2019a)	75.01/– –	87.09/90.12	62.48/– –	77.50/83.05
	MLMAN(Ye and Ling, 2019)	79.01/82.98	88.86/92.66	67.37/75.59	80.07/87.29
BERT	Proto-BERT (Han et al., 2018)	82.92/80.68	91.32/89.60	73.24/71.48	83.68/82.89
	MAML (Finn et al., 2017)	82.93/89.70	86.21/83.55	73.20/83.17	86.06/88.51
	GNN (Satorras and Estrach, 2018)	– –/75.66	– –/89.06	– –/70.08	– –/76.93
	BERT-PAIR (Gao et al., 2019b)	85.66/88.32	89.48/93.22	76.84/80.63	81.76/87.02
	REGRAB (Qu et al., 2020)	87.95/90.30	92.54/94.25	80.26/84.09	86.72/89.93
	TD-Proto (Yang et al., 2020)	– –/84.76	– –/92.38	– –/74.32	– –/85.92
	ConceptFERE (Yang et al., 2021)	– –/89.21	– –/90.34	– –/75.72	– –/81.82
	HCRP (Han et al., 2021a)	90.90/93.76	93.22/95.66	84.11/89.95	87.79/92.10
	SimpleFSRE (Liu et al., 2022)	91.29/94.42	94.05/96.37	86.09/90.73	89.68/93.47
	GM_GEN	**92.65/94.89**	**95.62/96.96**	**86.81/91.23**	**91.27/94.30**
BERT w/ P	MTB (Soares et al., 2019)	– –/91.10	– –/95.40	– –/84.30	– –/91.80
	CP (Peng et al., 2020)	– –/95.10	– –/97.10	– –/91.20	– –/94.70
	LDUR (Han et al., 2021b)	87.21/90.40	94.86/96.95	80.34/84.68	91.36/94.15
	HCRP (CP) (Han et al., 2021a)	94.10/96.42	96.05/97.96	89.13/93.97	93.10/96.46
	SimpleFSRE (CP) (Liu et al., 2022)	96.21/96.63	97.07/97.93	93.38/94.94	95.11/96.39
	GM_GEN (CP)	**96.97/97.03**	**98.32/98.34**	**93.97/94.99**	**96.58/96.91**

Table 1: Comparison results in terms of accuracy (%) for FSRE methods on FewRel 1.0 validation / test set.

Experiment

Model	5-1	5-5	10-1	10-5
Proto-CNN	35.09	49.37	22.98	35.22
Proto-BERT	40.12	51.5	26.45	36.93
Proto-PAIR	67.41	78.57	54.89	66.85
HCRP	76.34	83.03	63.77	72.94
GM_GEN	**76.67**	**91.28**	**64.19**	**84.84**

Table 2: Comparison results in terms of accuracy (%) for FSRE methods on FewRel 2.0 validation / test set.

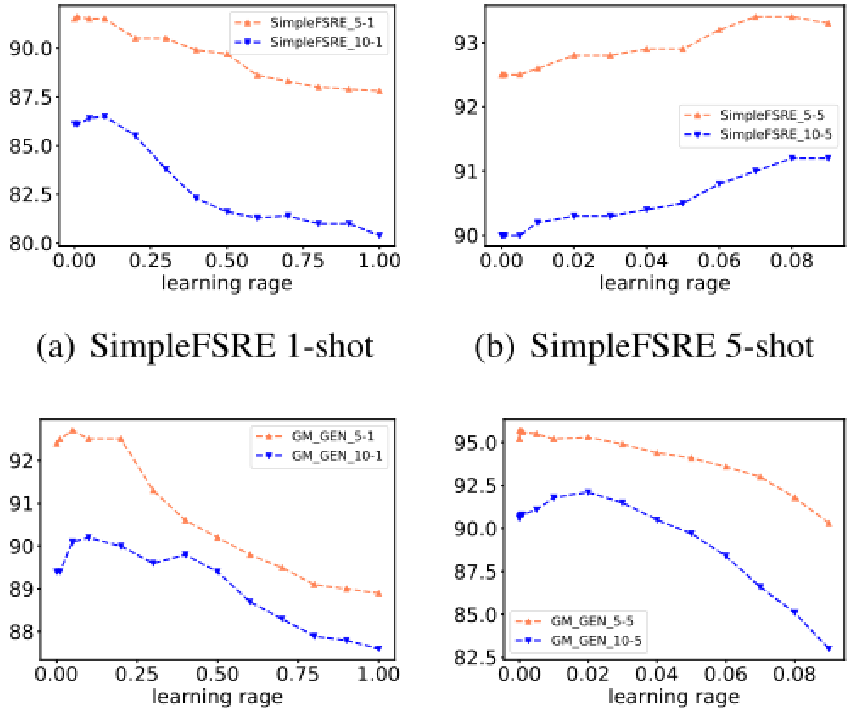
Model	5-1	5-5	10-1	10-5
GM_GEN	94.89	96.96	91.23	94.30
ADD_Base	94.46 (0.43↓)	96.18 (0.78↓)	88.91 (2.32↓)	93.43 (0.87↓)
Add_GEN	94.55 (0.34↓)	96.49 (0.47↓)	90.65 (0.58↓)	94.06 (0.24↓)
GM_CLS	94.76 (0.13↓)	96.53 (0.43↓)	91.06 (0.17↓)	93.61 (0.69↓)

Table 3: Ablation results in terms of accuracy (%) on FewRel 1.0 test set. ↓ denotes a drop of F1 score.

Experiment

	5-1		10-5	
	time	space	time	space
GM_GEN	5264.2S	111.8M	15533.4S	111.8M
GNN	5314.7S	110.3M	18335.9S	110.3M
SimpleFSRE	5256.1S	109.8M	15108.7S	109.8M
HCRP	5204.6S	110.7M	14573.3S	110.7M
Proto-BERT	4550.0S	109.5M	10612.2S	109.5M

Table 4: Complexity analysis. $S = Second$, $M = 1 \times 10^6$.

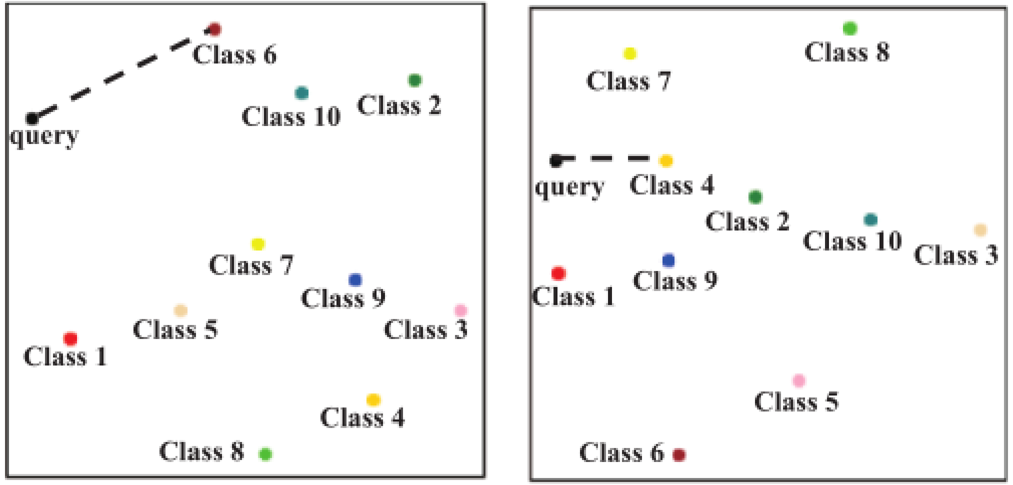


(a) SimpleFSRE 1-shot (b) SimpleFSRE 5-shot

(c) GM_GEN 1-shot (d) GM_GEN 5-shot

Figure 3: Impacts of the learning rate for the generated models on FewRel 1.0 validation set.

Experiment



(a) Before fine-tuning (b) After fine-tuning

Figure 4: Impacts of the fine-tuning on the generated models for FewRel 1.0 validation set. The dark dot represents the query sample, and other dots represent the prototypes of different classes. Note that the class 4 is the correct relation.

Thank you!