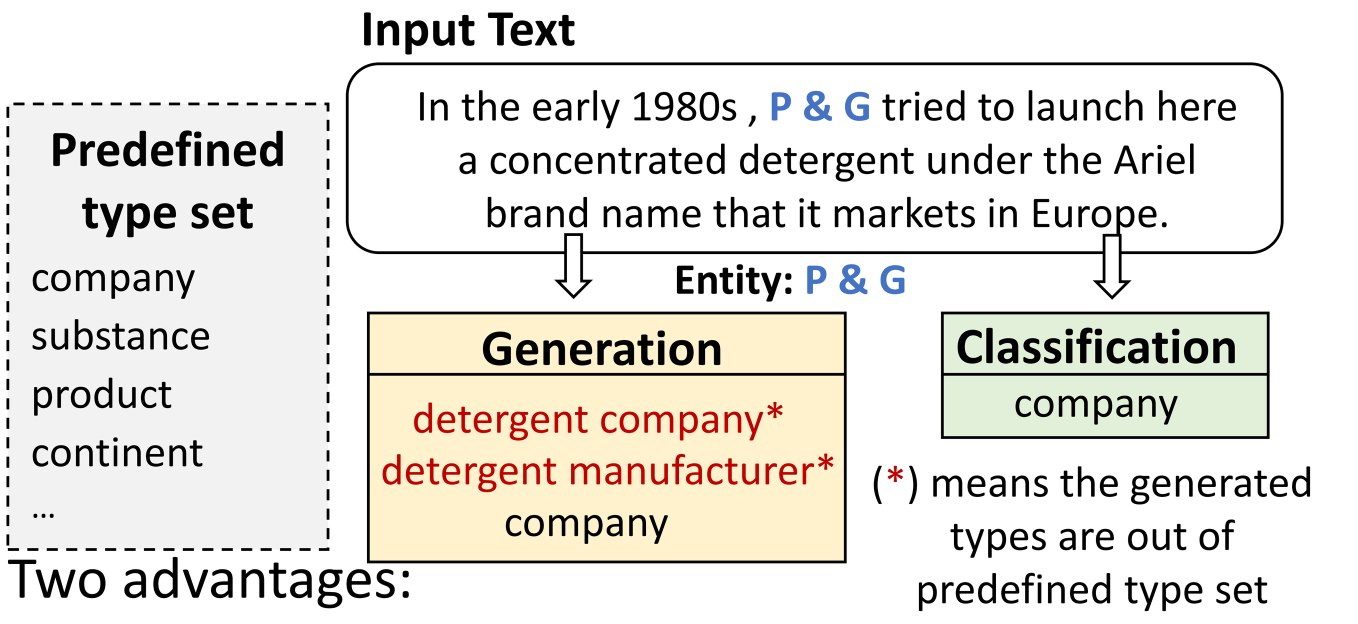
Generative Entity Typing with Curriculum Learning

Siyu Yuan, Deqing Yang*, Jiaqing Liang, Zhixu Li, Jinxi Liu,
Jingyue Huang, Yanghua Xiao*
Fudan University

Introduction

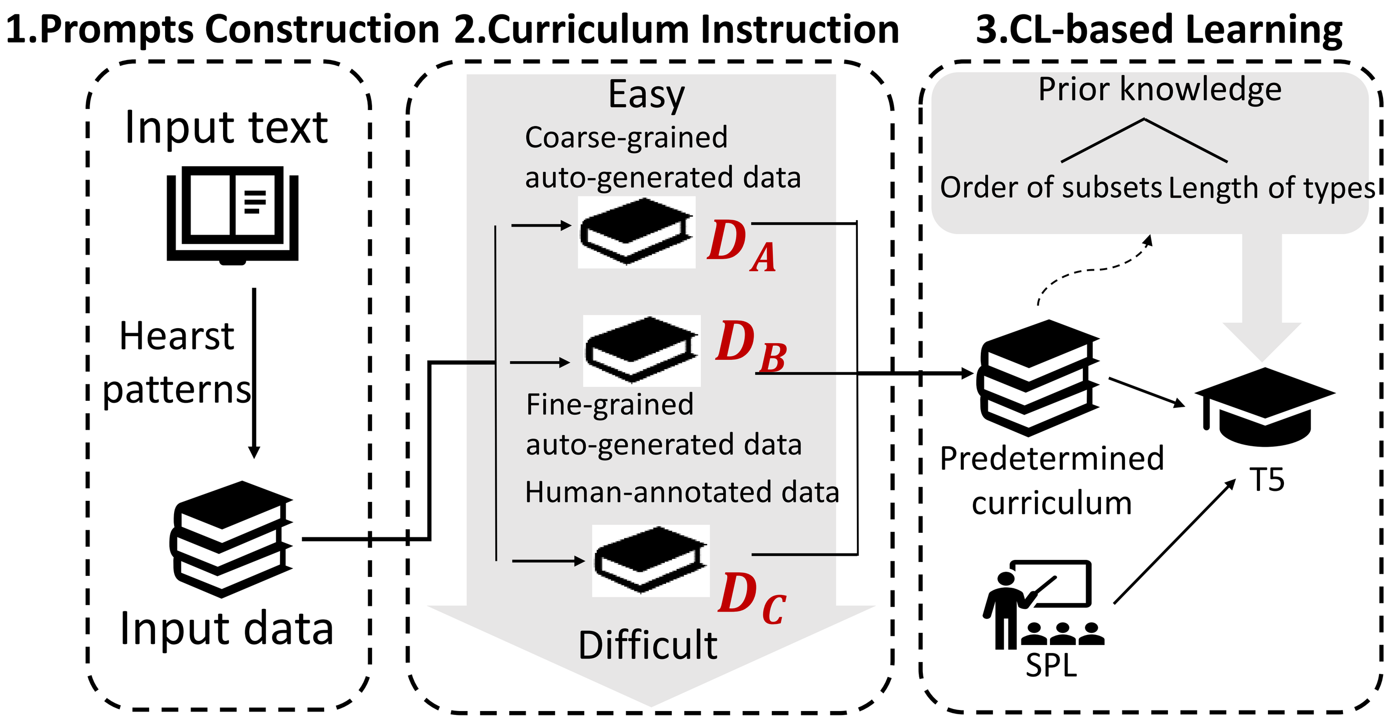
- **Task: Entity typing**
  - Aims to assign types to the entity mentions in given texts.
- **The drawbacks of pervious work**
  - Closed Type Set: Cannot assign the entity to the types out of the predefined set.
  - Few-shot Dilemma for Long-tail Types: Hardly handle few-shot and zero-shot issues.
- **Solution: Generative Entity Typing**
  - Generate types with a pre-trained language model from given a text with an entity mention.



- Two advantages:
  - Open Type Set: more open types for entity mentions
  - Conceptual Reasoning Capability: handle the few-shot and zero-shot dilemma well

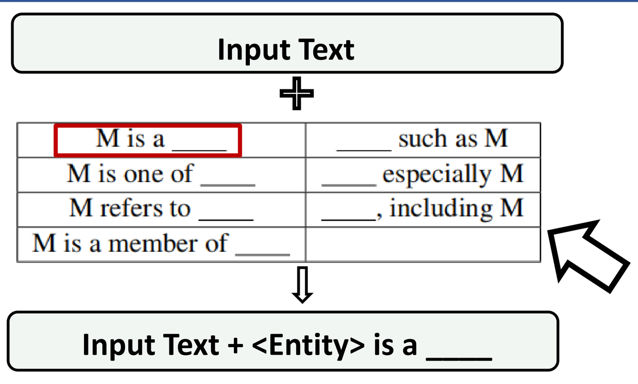
Challenges & Contribution

- **Challenge 1: Fine-grained Types Generation**
  - How to guide the PLMs to generate high-quality and fine-grained types for entities is crucial.
- **Challenge 2: Heterogeneous data**
  - How to train the PLMs to generate desirable types on these low-quality heterogeneous.
- **Solution: Curriculum Learning**
  - Better learn heterogeneous data by ordering the training samples based on their quality and difficulty.
  - CL can control the order of using training subsets from coarse-grained and lower-quality ones to fine-grained and higher-quality ones.



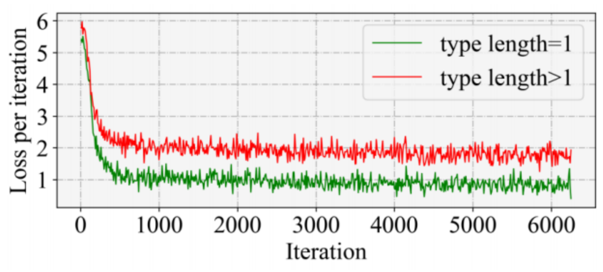
CL-based GET

- **Prompts Construction**
  - Construct the prompts in cloze format from the Hearst patterns



- **Curriculum Instruction**
  - It is more difficult for PLMs to fit the samples of fine-grained types.
  - Coarse-grained auto-generated data => *type length*=1
  - Fine-grained auto-generated data => *type length*≥2
  - Human-annotated data => *ultra fine−grained types*

Whole training data $D = D_A \cup D_B \cup D_C$



- **CL-based Learning (T5 backbone)**
  - A fixed curriculum ignores the feedback from the training process => adopt Self-paced Learning (SPL)
  - *Loss function of one sample* $D_k^{(i)} =< X^{(i)}, M^{(i)}, T_k^{(i)} > \Rightarrow L(D_k^{(i)}) = L_{CE}(T_k^{(i)}, f(X^{(i)}, \boldsymbol{\theta}, M^{(i)}))$
  - The training objective: $\min_{\boldsymbol{\theta},\boldsymbol{v}} E(\boldsymbol{\theta}, \boldsymbol{v}; \lambda) = \sum_{i=1}^{N} \sum_{k=1}^{K^{(i)}} v_k^{(i)} L(D_k^{(i)}) + g(\boldsymbol{v}; \lambda)$
  - The binary variable $v_k^{(i)} \in [0,1]$ $\lambda = \mu\lambda, \mu > 1$

$$v_k^{(i)} = \begin{cases} 1, & L(D_k^{(i)}) < \lambda \\ 0, & L(D_k^{(i)}) \geq \lambda \end{cases} \quad g(\boldsymbol{v}; \lambda) = -\sum_{i=1}^{N} \sum_{k=1}^{K^{(i)}} v_k^{(i)} \Rightarrow$$

More samples with larger losses are gradually incorporated

- Expect the model to be trained according to the predetermined curriculum => adopt prior knowledge

$$\gamma\left(D_k^{(i)}\right) = \begin{cases} 1, if\, D_k^{(i)} \in D_A \\ 2, if\, D_k^{(i)} \in D_B \\ 3, if\, D_k^{(i)} \in D_C \end{cases} + \; length(T_k^{(i)}) \Rightarrow w(D_k^{(i)}) = length(T_k^{(i)}) \times \gamma(D_k^{(i)})$$

$$\Downarrow$$

$$L(D_k^{(i)}) = L_{CE}(T_k^{(i)}, f(X^{(i)}, \boldsymbol{\theta}, M^{(i)})) \times w(D_k^{(i)})$$

Order of subsets   Length of types

A sample with a large weight would be incorporated later

Experiment

- **Datasets: 4 dataset**
- **Metrics:**
  - CT # ⇨ The number of correct types
  - Len. ⇨ The average length of types
  - Precision, Relative Recall, Relative F1

Dataset	Type	Language	Size of D3	Size of Test set
BNN (Weischedel and Brunstein, 2005)	Coarse-grained	English	10,000	500
FIGER (Shimaoka et al., 2016)	Fine-grained	English	10,000	278
Ultra-Fine (Choi et al., 2018)	Ultra fine-grained	English	5500	500
GT (Lee et al., 2020)	Multilingual	English	4,750	250
		Chinese	4,750	250

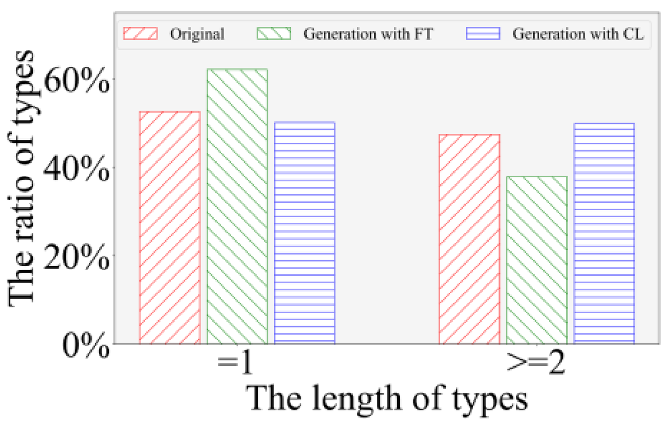
- **RQ1: Can GET improve the task performance? A1: Yes**

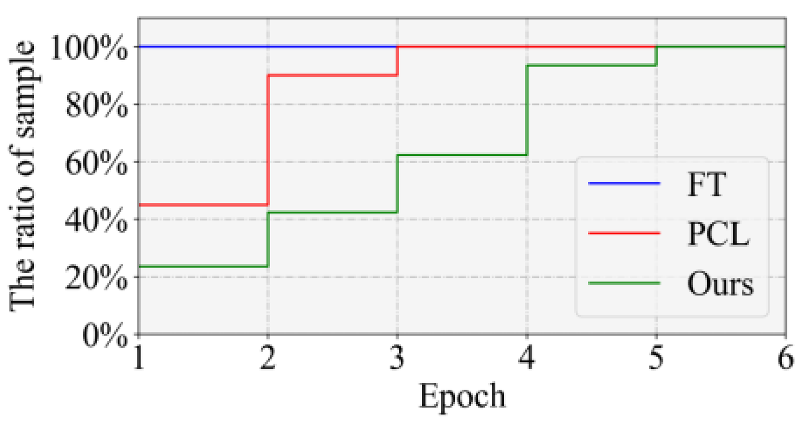
Model	BNN				FIGER			
	CT #	Prec.	R-Recall	R-F1	CT #	Prec.	R-Recall	R-F1
Zhang et al. (2018)	555	58.10%	50.49%	54.03%	348	62.00%	49.85%	55.26%
Lin and Ji (2019)	534	55.90%	48.58%	51.98%	353	62.90%	50.57%	56.07%
Xiong et al. (2019)	558	58.40%	50.75%	54.31%	350	62.30%	50.09%	55.53%
Ali et al. (2020)	697	73.00%	63.43%	67.88%	399	**71.00%**	57.08%	63.29%
Chen et al. (2020)	718	75.20%	65.35%	69.93%	388	69.10%	55.56%	61.59%
Zhang et al. (2021)	732	76.70%	66.65%	71.32%	394	70.10%	56.36%	62.48%
Li et al. (2021)	668	69.90%	60.74%	65.00%	397	70.60%	56.76%	62.93%
Ours	**875**	**82.30%**	**79.62%**	**80.94%**	**444**	66.20%	**63.52%**	**64.83%**

Model	Ultra-Fine			
	CT #	Prec.	R-Recall	R-F1
Xiong et al. (2019)	782	50.30%	24.28%	32.75%
Onoe and Durrett (2019) ELMo	884	51.50%	27.44%	35.81%
Onoe and Durrett (2019) BERT	884	51.60%	27.44%	35.83%
López and Strube (2020)	915	43.40%	28.41%	34.34%
Onoe et al. (2021)	1039	52.80%	32.26%	40.05%
Liu et al. (2021b)	1042	54.50%	32.35%	40.60%
Dai et al. (2021)	1213	53.60%	37.66%	44.24%
Ours	**1275**	**87.10%**	**39.58%**	**54.43%**

- **RQ2: How do CL improve GET performance?**
  - Let the model pay more attention to fine-grained types.
  - the former subsets can be regarded as a pre-training process that helps model optimization and regularizes the training on the later subsets.

Model	Dataset	Chinese				English			
		CT #	Prec.	R-F1	Len.	CT #	Prec.	R-F1	Len.
FT	Auto-generated data	690	84.46%	70.81%	2.80	870	75.85%	52.87%	1.48
PCL		646	91.76%	70.37%	2.75	864	85.97%	54.87%	1.32
SPL w/o PK		672	**92.18%**	72.22%	2.75	900	87.12%	56.66%	1.54
Ours		**714**	90.04%	**74.18%**	**2.86**	**928**	**87.14%**	**57.84%**	**1.62**
FT	Human-annotated data	383	72.54%	53.98%	**2.65**	352	84.82%	48.82%	1.72
PCL		370	77.24%	54.01%	2.64	**375**	88.03%	51.62%	1.69
SPL w/o PK		383	78.64%	55.59%	2.61	370	90.46%	51.53%	1.74
Ours		**409**	**83.64%**	**59.28%**	2.63	373	**90.75%**	**51.88%**	**1.82**





- **RQ3: How do we adopt the typing generated from our GET model?**
  - Short Text Classification and Entity linking

Method	Prec.	Recall	F1
No type	72.92%	72.70%	72.47%
types (KG)	73.99%	73.17%	73.30%
types (Gen.)	**74.51%**	**73.41%**	**73.53%**

Dataset	Method	F1
AIDA CoNLL-YAGO	triples (KG.)	94.58%
	triples (Gen.)	**94.92%**
ACE 2014	triples (KG)	89.74%
	triples (Gen.)	**90.54%**

If you have any questions, please email: syyuan21@m.fudan.edu.cn