EMNLP 2022: Generative Entity Typing with Curriculum Learning

Siyu Yuan

syyuan21@m.fudan.edu.cn

KW@Fudan

Outline

- Background
- Generative Entity Typing
- GET with Curriculum Learning
- Experiments
- Takeaways

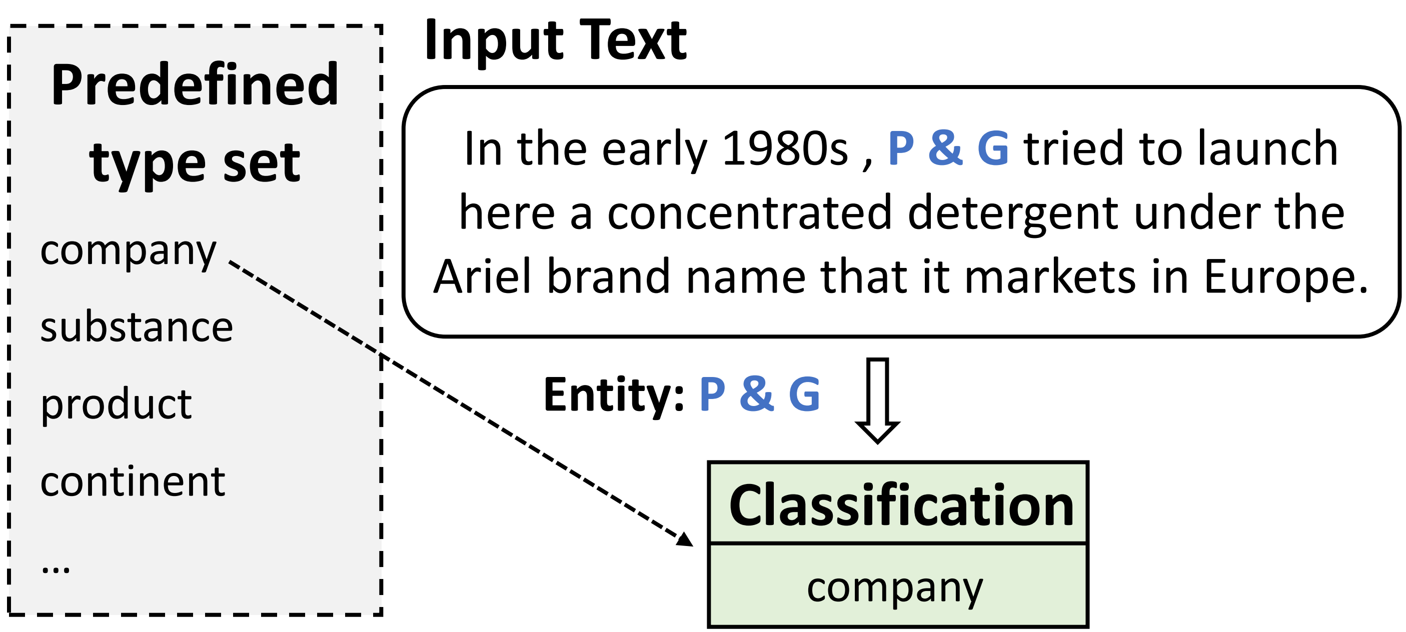
Entity Typing

知識工場

- *Entity typing aims to assign types to the entity mentions in given texts.*

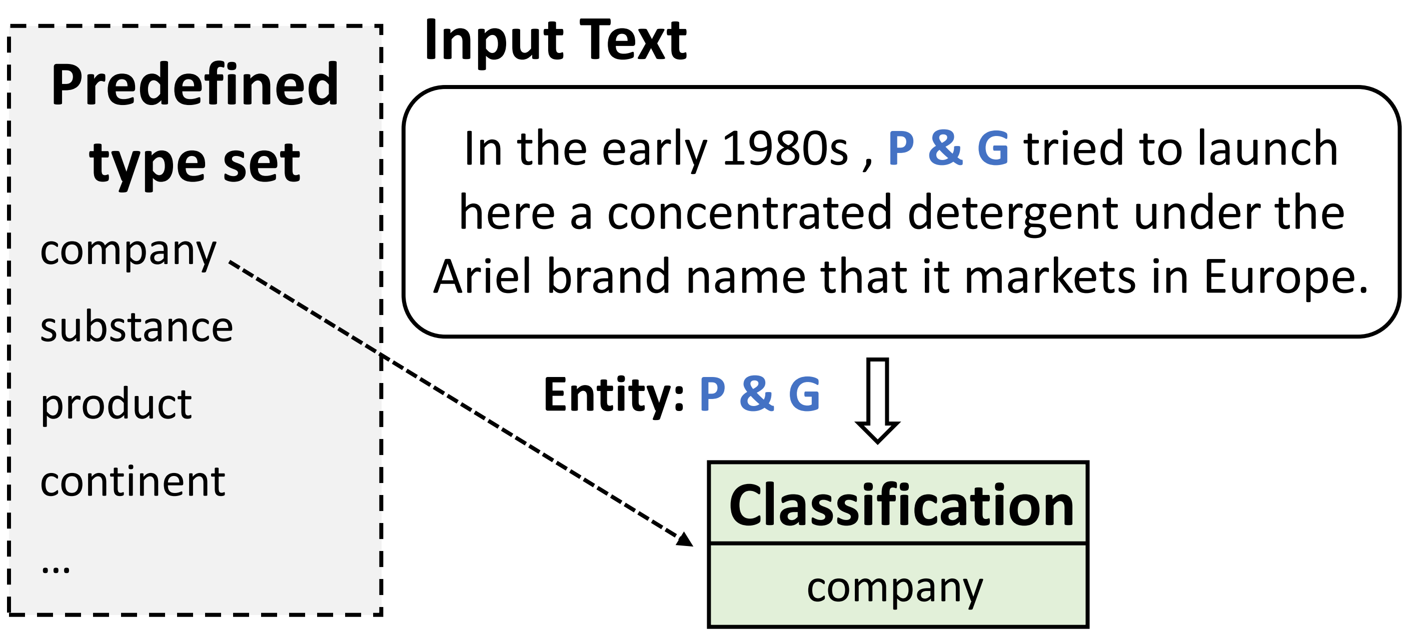
Entity Typing

- *Entity typing aims to assign types to the entity mentions in given texts.*



Entity Typing

- *Entity typing aims to assign types to the entity mentions in given texts.*



Classification Paradigm

Drawbacks

- *Closed Type Set*

Predefined type set		
company	product	…
substance	continent	

⇦ *Limited*

Cannot assign the entity to the types
out of the predefined set

Drawbacks

- *Few-shot Dilemma for Long-tail Types*
  - *Hardly handle few-shot and zero-shot issues*
    - *more than 80% types have less than 5 instances*
    - *25% types even never appear in the training data from the ultra-fine dataset*

Generative Entity Typing

- *Definition:*
  - *Generate types with a pre-trained language model from given a text with an entity mention*

Generative Entity Typing

知識工場

- *Definition:*
  - *Generate types with a pre-trained language model from given a text with an entity mention*

Open Type Set

⇩

Generate more open types for entity mentions

Generative Entity Typing

知識工場

- *Definition:*
  - *Generate types with a pre-trained language model from given a text with an entity mention*

Open Type Set

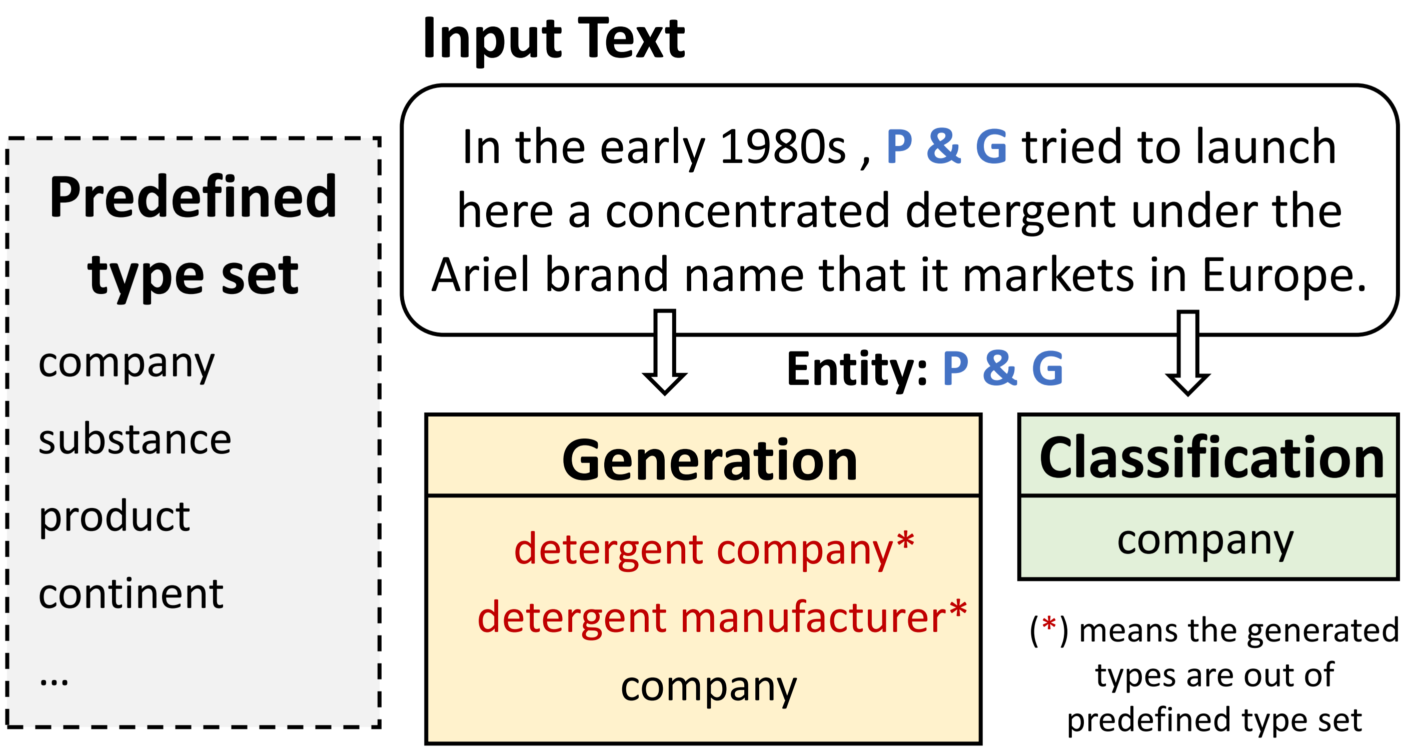
Generate more open types for entity mentions

Conceptual Reasoning Capability

Handle the few-shot and zero-shot dilemma well

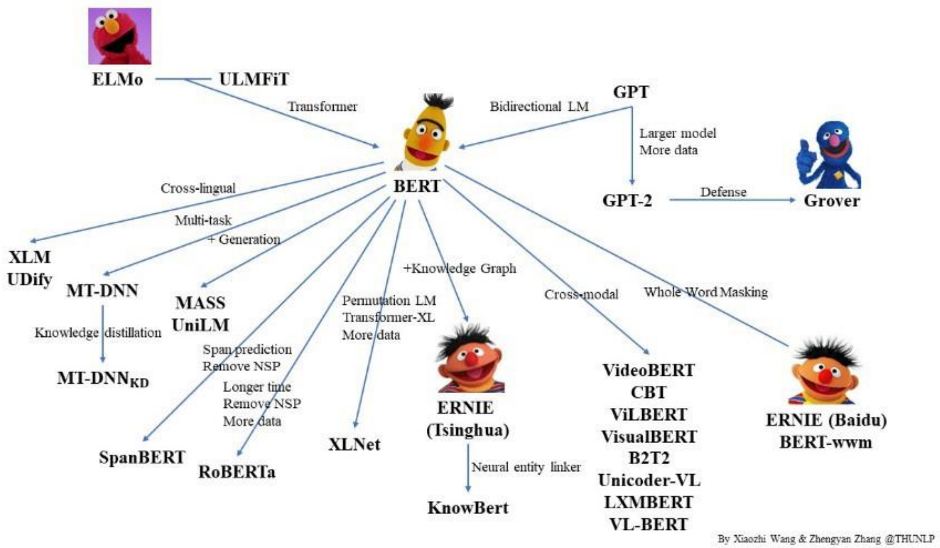
Generative Entity Typing

- *Definition:*
  - *Generate types with a pre-trained language model from given a text with an entity mention*



Challenges

🤔 *Challenge 1: Fine-grained Types Generation*



Biased to generate
high-frequency vocabulary

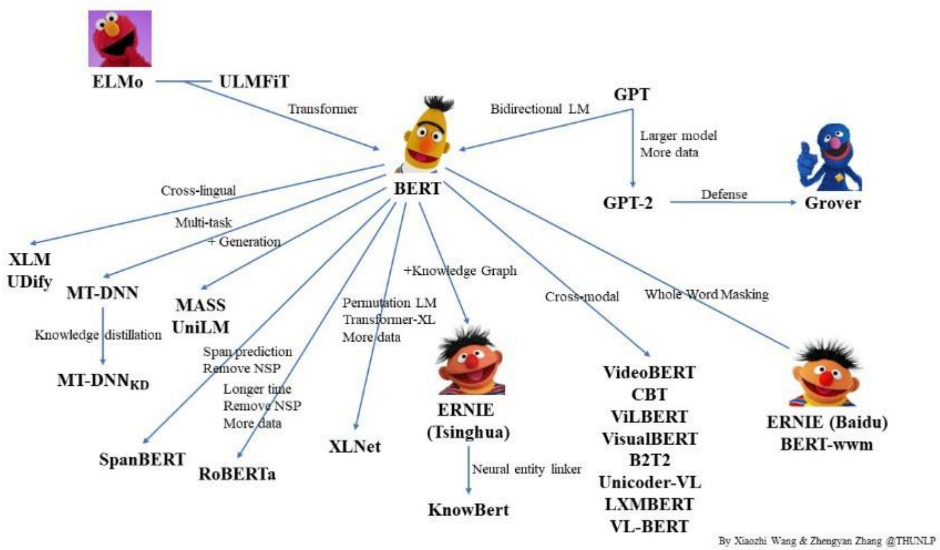
⇩

Tends to generate
coarse-grained types

Challenges

知識
工場

Challenge 1: Fine-grained Types Generation



Biased to generate
high-frequency vocabulary

⇩

Tends to generate
coarse-grained types

How to guide the PLMs to generate high-quality and fine-grained types for entities is crucial.

Challenges

知識工場

🤔 *Challenge 2: heterogeneous data*

Ultra fine-grained entity typing dataset

Human-annotated data
- *Less than 10%*
- *High-quality*

Auto-generated data
- *More than 90%*
- *Low-quality*

知識
工場

Challenges

Challenge 2: heterogeneous data

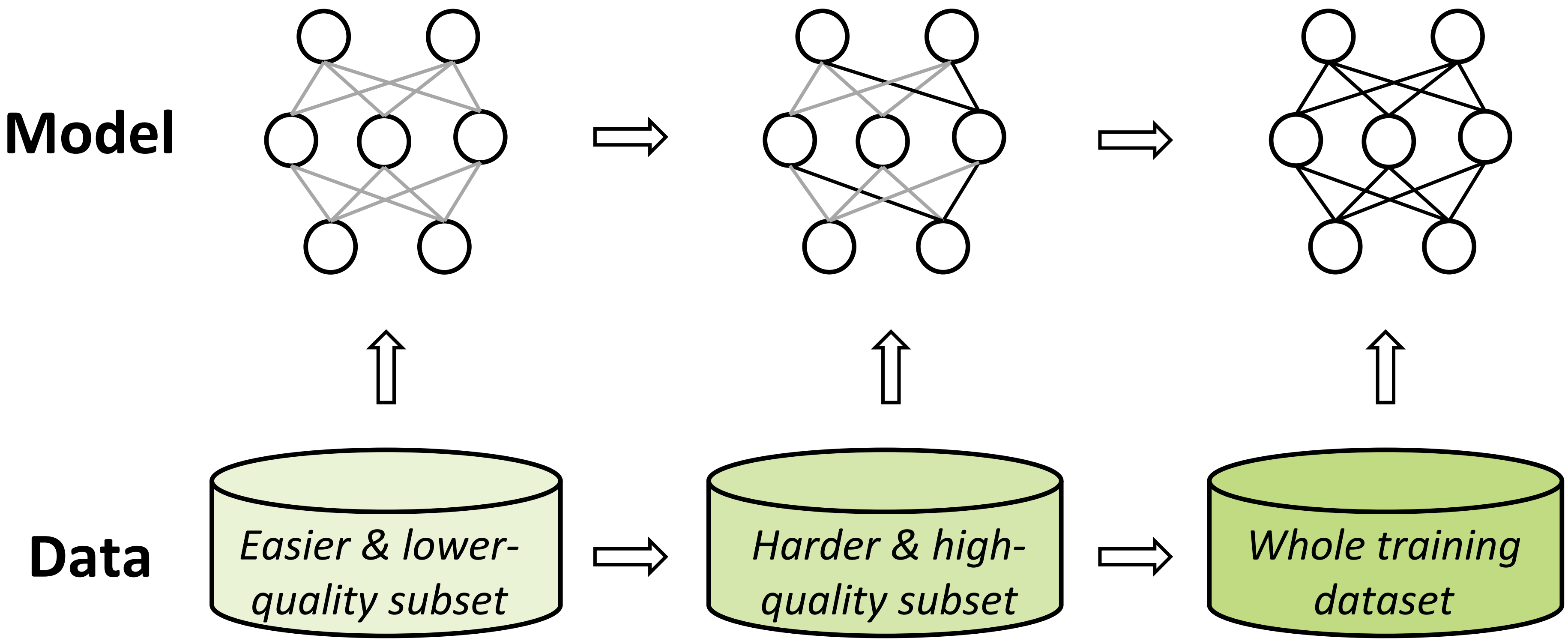
Ultra fine-grained entity typing dataset

Human-annotated data
- *Less than 10%*
- *High-quality*

Auto-generated data
- *More than 90%*
- *Low-quality*

How to train the PLMs to generate desirable types on these low-quality heterogeneous.

Curriculum Learning

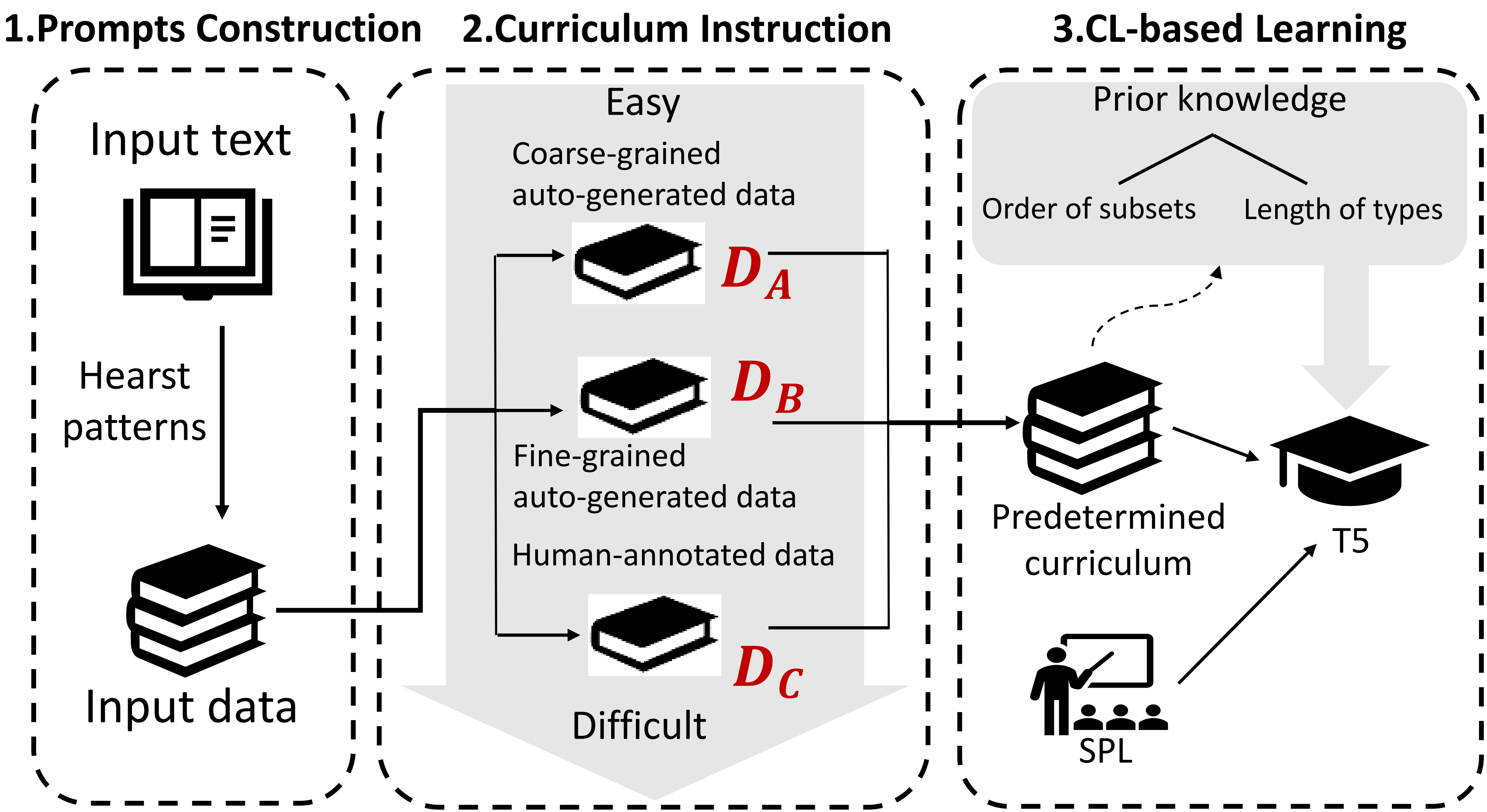


Curriculum Instruction

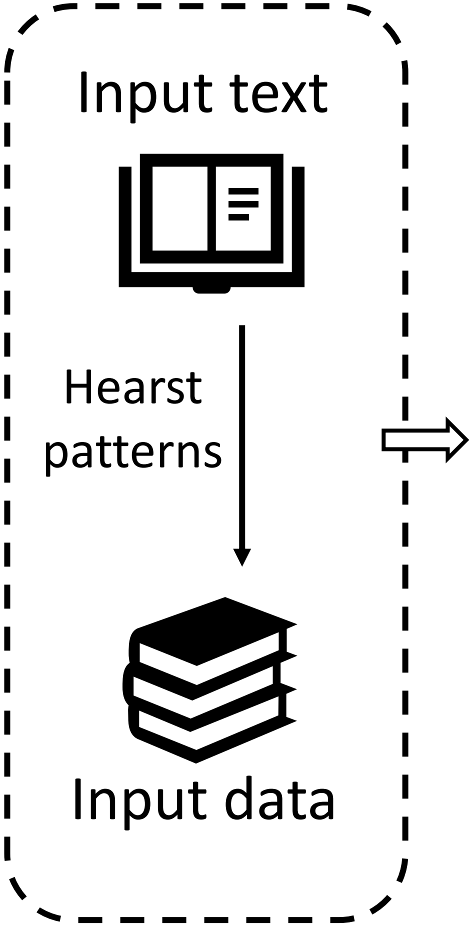
CL-based GET : Overview

A CL-based strategy to train GET model

CL-based GET : Overview

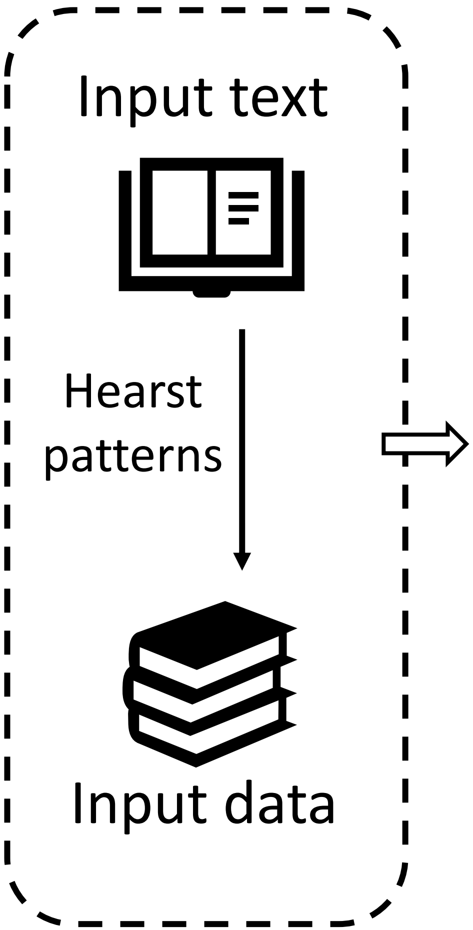


Prompts Construction



M is a ____	____ such as M
M is one of ____	____ especially M
M refers to ____	____, including M
M is a member of ____	

Prompts Construction



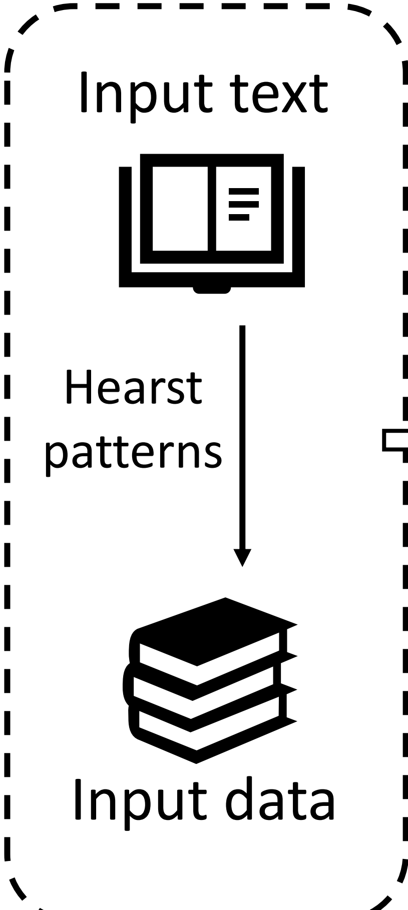
Input Text

In the early 1980s , **P & G** tried to launch here a concentrated detergent under the Ariel brand name that it markets in Europe.

+

M is a ____	____ such as M
M is one of ____	____ especially M
M refers to ____	____, including M
M is a member of ____	

Prompts Construction



Input Text

In the early 1980s , P & G tried to launch here a concentrated detergent under the Ariel brand name that it markets in Europe.

+

M is a ____	____ such as M
M is one of ____	____ especially M
M refers to ____	____, including M
M is a member of ____	

Input Data

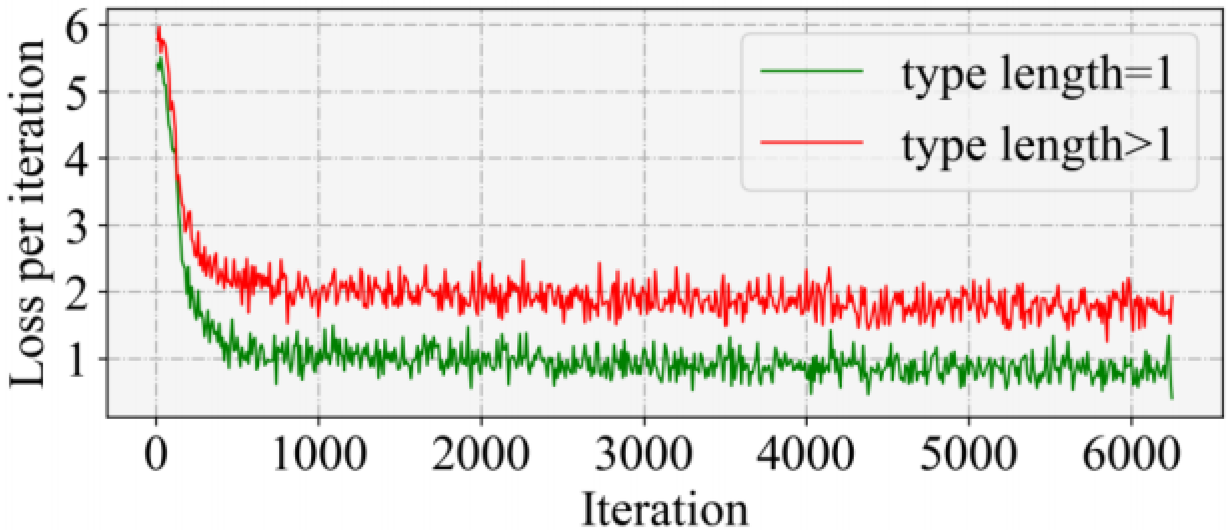
In the early 1980s , P & G tried to launch here a concentrated detergent under the Ariel brand name that it markets in Europe.
P & G is a ____

Curriculum Instruction

Does the difficulty of a sample for model learning greatly depend on the granularity of its type?

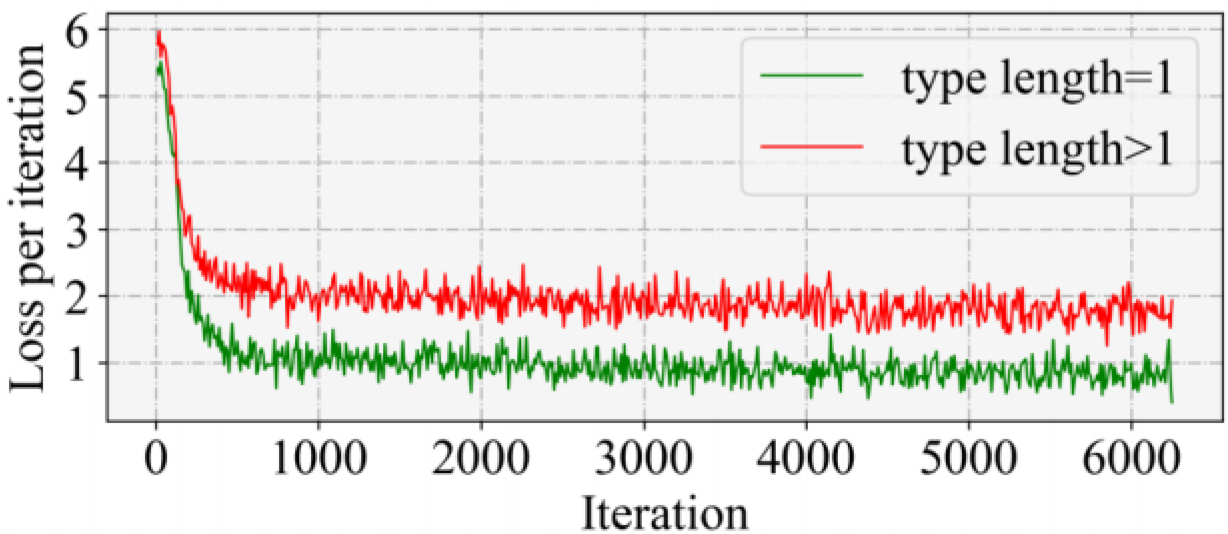
Curriculum Instruction

Does the difficulty of a sample for model learning greatly depend on the granularity of its type?



Curriculum Instruction

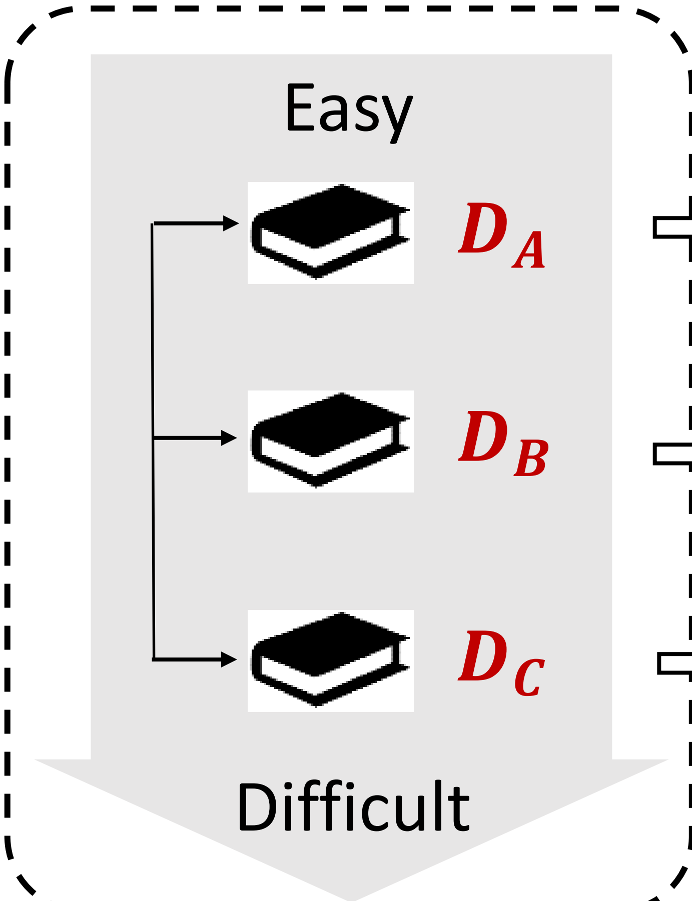
Does the difficulty of a sample for model learning greatly depend on the granularity of its type?



It is more difficult for PLMs to fit the training samples of fine-grained types.

Curriculum Instruction

2.Curriculum Instruction



Coarse-grained auto-generated data

$type\ length = 1$

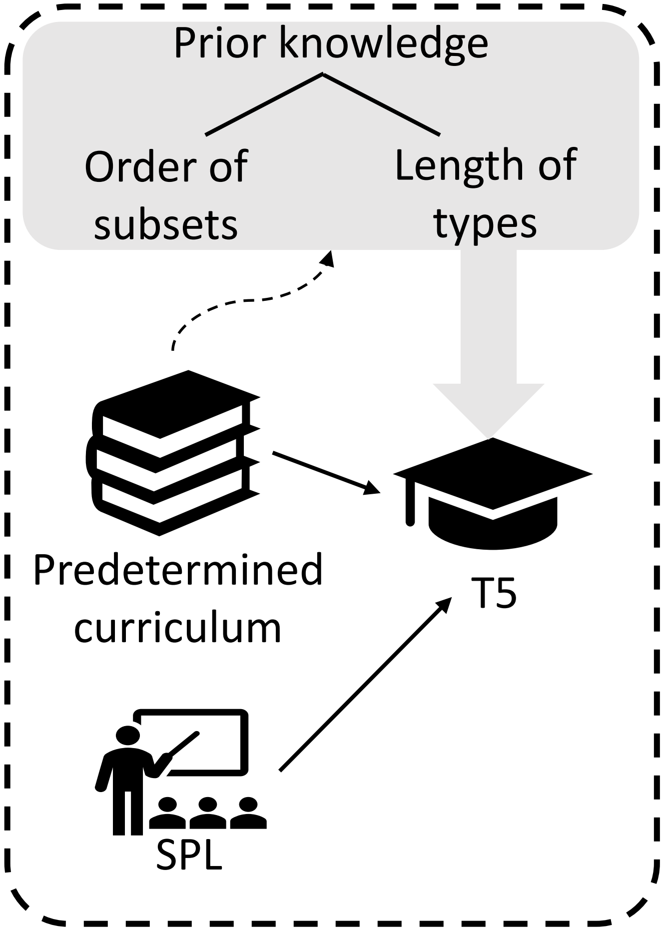
Fine-grained auto-generated data

$type\ length \geq 2$

Human-annotated data

$ultra\ fine-grained\ types$

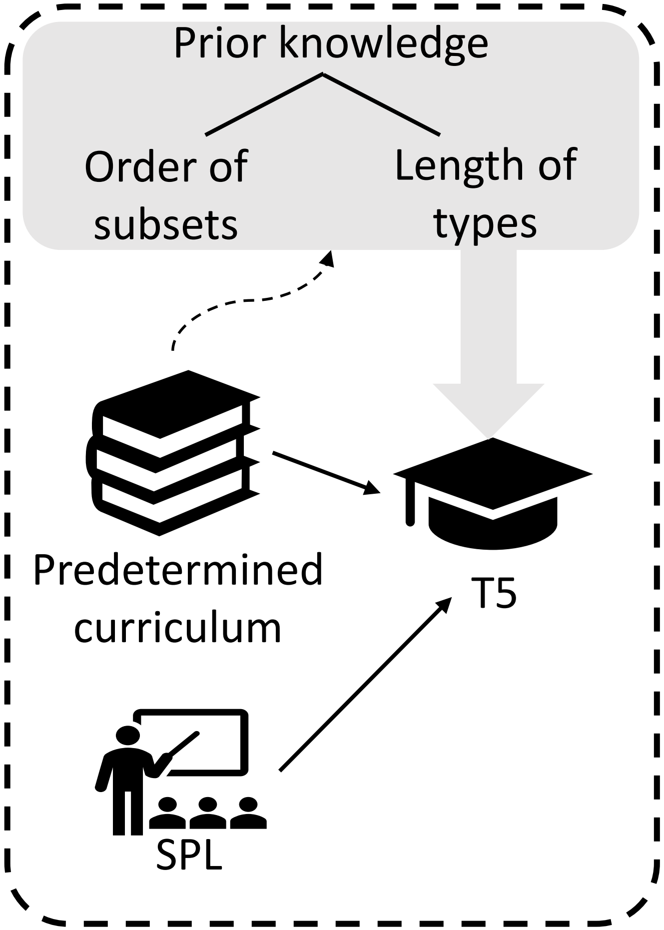
CL-based Learning



T5 backbone

An encoder-decoder pre-trained model

CL-based Learning



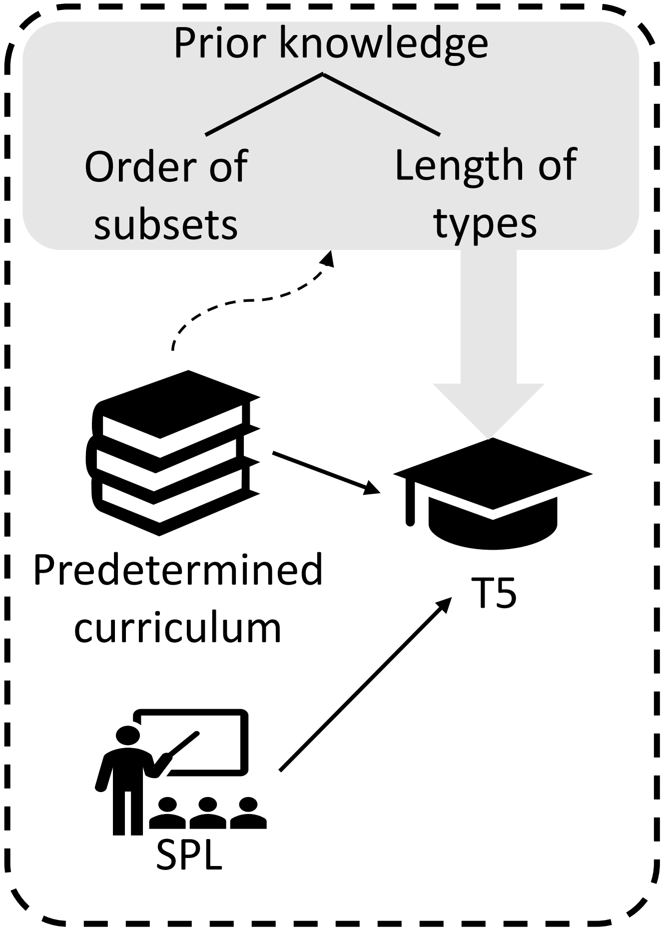
T5 backbone

An encoder-decoder pre-trained model

Loss function of one sample $\boldsymbol{D_k^{(i)}}$

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

CL-based Learning



T5 backbone

An encoder-decoder pre-trained model

Loss function of one sample $\boldsymbol{D_k^{(i)}}$

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

$$\Uparrow$$

$$D_k^{(i)} =< X^{(i)}, M^{(i)}, T_k^{(i)} >$$

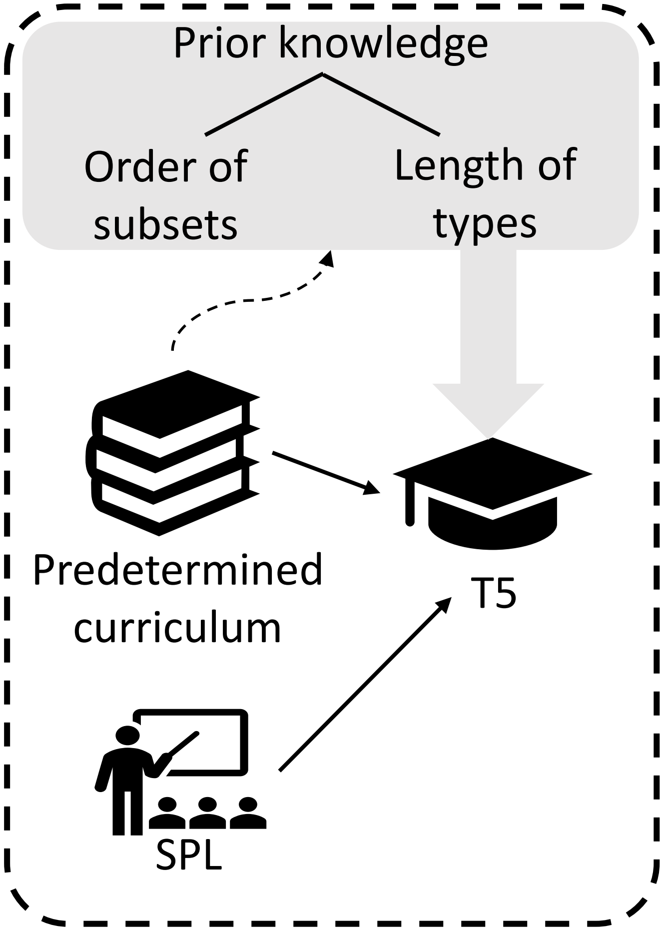
$$\Uparrow$$

Whole training data

$D = D_A \cup D_B \cup D_C$

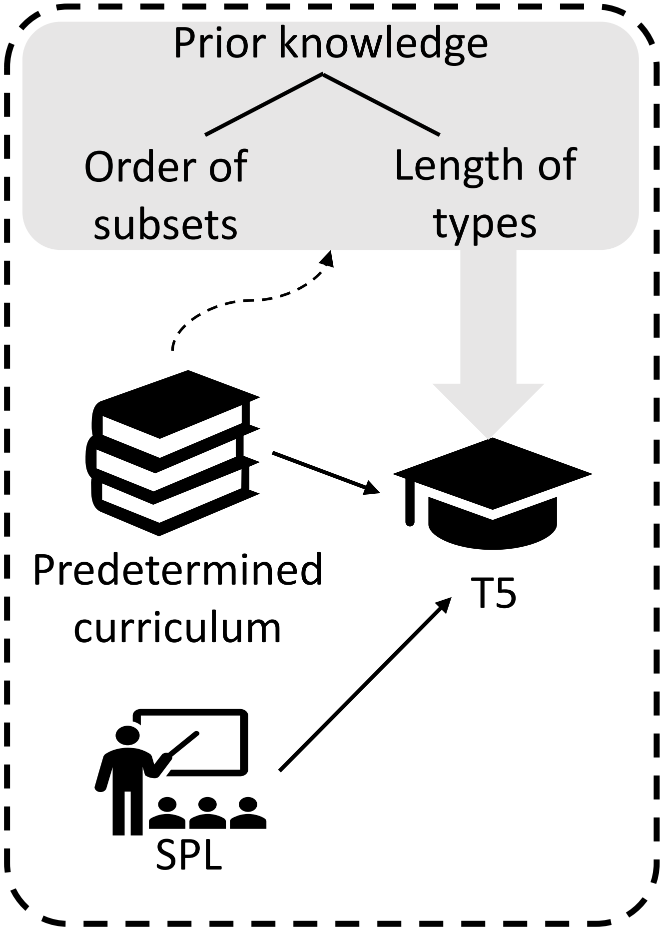
A type for entity mention $M^{(i)}$ w.r.t. the context of $X^{(i)}$

CL-based Learning



Why do we adopt Self-paced Learning (SPL)?

CL-based Learning



Why do we adopt Self-paced Learning (SPL)?

⇩

A fixed curriculum ignores the feedback from the training process

SPL-based Training Process

知識工場

- ***The training objective***

$$\min_{\boldsymbol{\theta},\boldsymbol{v}} E(\boldsymbol{\theta}, \boldsymbol{v}; \lambda) = \sum_{i=1}^{N} \sum_{k=1}^{K^{(i)}} v_k^{(i)} L(D_k^{(i)}) + \mathrm{g}(\boldsymbol{v}; \lambda)$$

SPL-based Training Process

知識工場

- ***The training objective***

$$\min_{\boldsymbol{\theta},\boldsymbol{\nu}} E(\boldsymbol{\theta},\boldsymbol{\nu};\lambda) = \sum_{i=1}^{N}\sum_{k=1}^{K^{(i)}} v_k^{(i)} L(D_k^{(i)}) + \mathrm{g}(\boldsymbol{\nu};\lambda)$$

- ***The binary variable*** $\boldsymbol{v}_k^{(i)} \in [\mathbf{0},\mathbf{1}]$

$$v_k^{(i)} = \begin{cases} 1, & L(D_k^{(i)}) < \lambda \\ 0, & L(D_k^{(i)}) \geq \lambda \end{cases} \Longleftarrow$$

Whether the sample $D_k^{(i)}$ should be considered

SPL-based Training Process

知識工場

- ***The training objective***

$$\min_{\boldsymbol{\theta},\boldsymbol{\nu}} E(\boldsymbol{\theta},\boldsymbol{\nu};\lambda) = \sum_{i=1}^{N}\sum_{k=1}^{K^{(i)}} v_k^{(i)} L(D_k^{(i)}) + \mathrm{g}(\boldsymbol{\nu};\lambda)$$

- ***The binary variable*** $\boldsymbol{v_k^{(i)}} \in [\mathbf{0},\mathbf{1}]$

$$v_k^{(i)} = \begin{cases} 1, & L(D_k^{(i)}) < \lambda \\ 0, & L(D_k^{(i)}) \geq \lambda \end{cases} \Longleftarrow$$

Whether the sample $D_k^{(i)}$ *should be considered*

- ***Binary self-paced function***

$$g(\boldsymbol{\nu};\lambda) = -\sum_{i=1}^{N}\sum_{k=1}^{K^{(i)}} v_k^{(i)} \Longleftarrow$$

A regularizer to avoid over-fitting

SPL-based Training Process

知識
工場

- ***The training objective***

$$\min_{\boldsymbol{\theta},\boldsymbol{v}} E(\boldsymbol{\theta},\boldsymbol{v};\lambda) = \sum_{i=1}^{N}\sum_{k=1}^{K^{(i)}} v_k^{(i)} L(D_k^{(i)}) + \mathrm{g}(\boldsymbol{v};\lambda)$$

- ***The binary variable*** $\boldsymbol{v}_{\boldsymbol{k}}^{(\boldsymbol{i})} \in [\mathbf{0},\mathbf{1}]$

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

$\Rightarrow$ *"Easy" samples with small losses are first used for training*

SPL-based Training Process

- ***The training objective***

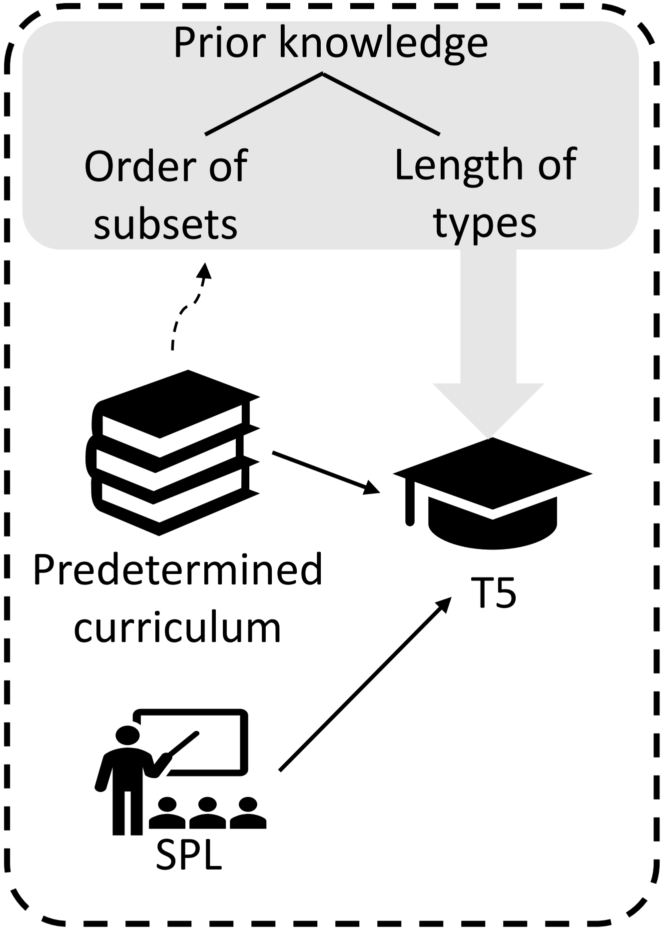
$$\min_{\boldsymbol{\theta},\boldsymbol{\nu}} E(\boldsymbol{\theta},\boldsymbol{\nu};\lambda) = \sum_{i=1}^{N}\sum_{k=1}^{K^{(i)}} v_k^{(i)} L(D_k^{(i)}) + \mathrm{g}(\boldsymbol{\nu};\lambda)$$

- ***The binary variable*** $\boldsymbol{v_k^{(i)}} \in [\mathbf{0},\mathbf{1}]$

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

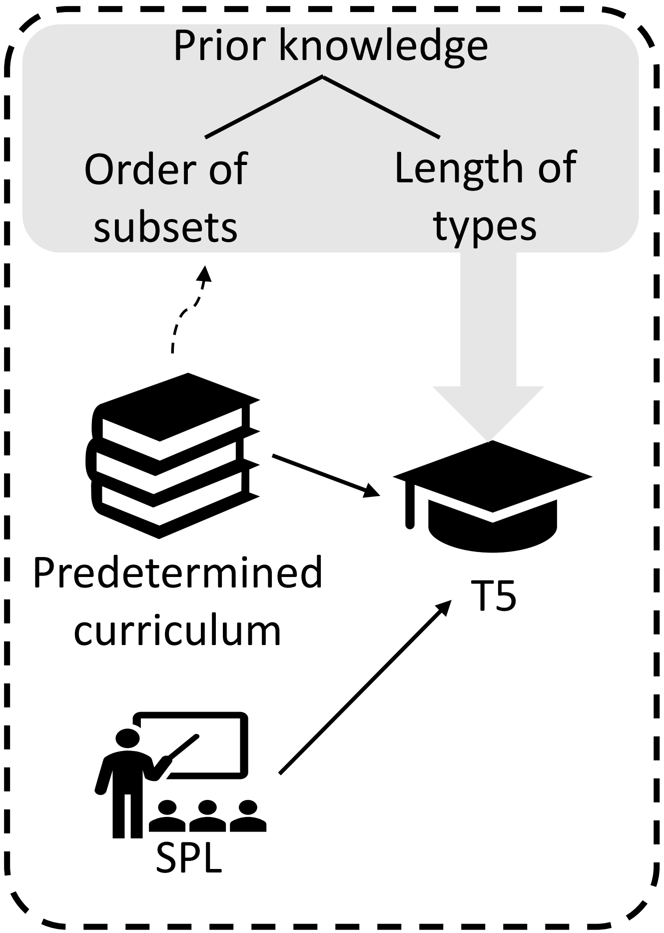
$\lambda = \mu\lambda, \mu > 1 \Longrightarrow$ *More samples with larger losses are gradually incorporated*

CL-based Learning



Why do we adopt prior knowledge?

CL-based Learning

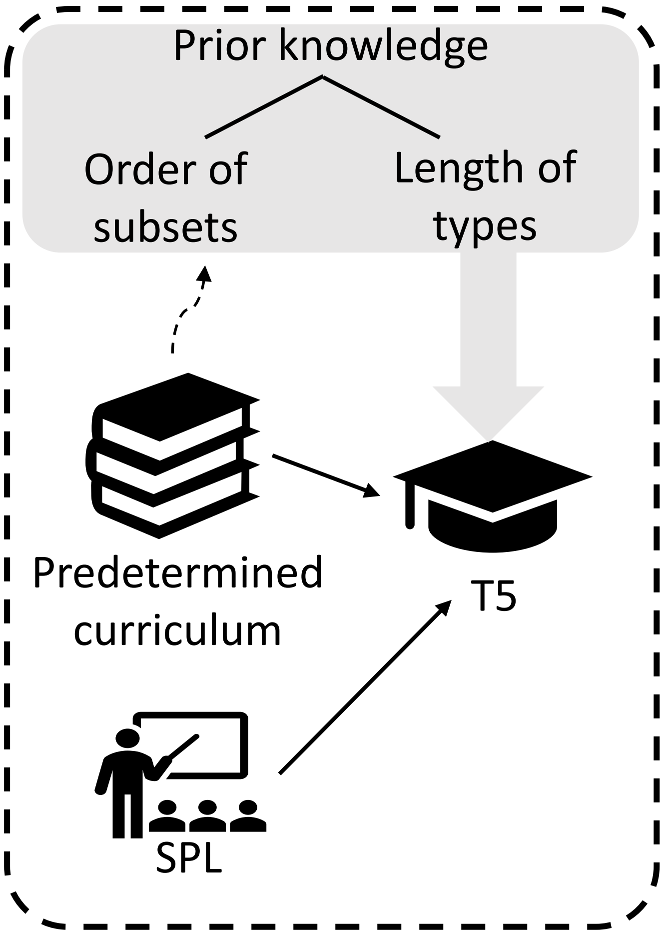


Why do we adopt prior knowledge?

⇩

Expect the model to be trained according to the predetermined curriculum

CL-based Learning



Why do we adopt prior knowledge?

⇩

Expect the model to be trained according to the predetermined curriculum

Intervene SPL

Prior Knowledge

Prior knowledge

Order of subsets

Length of types

Prior Knowledge

Prior knowledge

Order of subsets　　Length of types

$$\gamma\left(D_k^{(i)}\right) = \begin{cases} 1, if\, D_k^{(i)} \in D_A \\ 2, if\, D_k^{(i)} \in D_B \\ 3, if\, D_k^{(i)} \in D_C \end{cases}$$

Prior Knowledge

知識工場

Prior knowledge

Order of subsets | Length of types

$$\gamma\left(D_k^{(i)}\right) = \begin{cases} 1, if\, D_k^{(i)} \in D_A \\ 2, if\, D_k^{(i)} \in D_B \\ 3, if\, D_k^{(i)} \in D_C \end{cases}$$

$$length(T_k^{(i)})$$

Prior Knowledge

Prior knowledge

Order of subsets   Length of types

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

$$\Downarrow$$

$$w(D_k^{(i)}) = length(T_k^{(i)}) \times \gamma(D_k^{(i)})$$

知識工場

Prior Knowledge

Prior knowledge

Order of subsets Length of types

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

$$w(D_k^{(i)}) = length(T_k^{(i)}) \times \gamma(D_k^{(i)})$$

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

Prior Knowledge

Prior knowledge

Order of subsets — Length of types

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

$$\Downarrow$$

$$w(D_k^{(i)}) = length(T_k^{(i)}) \times \gamma(D_k^{(i)})$$

$$\Downarrow$$

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

A sample with a large weight would be incorporated later by the training process

Experiments

- Dataset:

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

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

Overall Results

知識
工場

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%**

Table 3: Comparison results of different approaches on the sample test set in coarse-grained and fine-grained entity typing dataset.

- *Our model significantly improves precision and covers more entity types*

Overall Results

知識工場

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%**

Table 4: Comparison results of different approaches on the sample test set in Ultra-fine entity typing dataset.

- *The classification-based approaches are **extremely difficult** to select the appropriate types from the large predefined type set.*
- *our GET model has **no classification constraint** since it transforms multi-classification into a generation paradigm that is more suitable for PLMs*

Overall Results

Dataset	MaNew	MiNew	R.New
BNN	4	100	11.61%
FIGER	25	137	26.81%
Ultra-Fine	73	543	42.14%

Table 5: The number and ratio of new types generated by our model on different datasets.

Total number of generated types beyond the predefined type set

Overall Results

Dataset	MaNew	MiNew	R.New
BNN	4	100	11.61%
FIGER	25	137	26.81%
Ultra-Fine	73	543	42.14%

Table 5: The number and ratio of new types generated by our model on different datasets.

Total number of generated types beyond the human-annotated type set of each instance

Overall Results

Dataset	MaNew	MiNew	R.New
BNN	4	100	11.61%
FIGER	25	137	26.81%
Ultra-Fine	73	543	42.14%

Table 5: The number and ratio of new types generated by our model on different datasets.

Ratio of new generated types per sample

Overall Results

Dataset	MaNew	MiNew	R.New
BNN	4	100	11.61%
FIGER	25	137	26.81%
Ultra-Fine	73	543	42.14%

Table 5: The number and ratio of new types generated by our model on different datasets.

- *Our model can generate abundant types that are not in the golden labeled set*

Effectiveness of CL

知識
工場

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**

Table 6: Performance comparisons of our model and its variants on the auto-generated and human-annotated test set.

- *The superiority of PCL and SPL over FT verifies **CL's advantage** over the general training strategy*
- *SPL w/o PK's superiority over PCL verifies **SPL**'s effectiveness*

Analysis of SPL

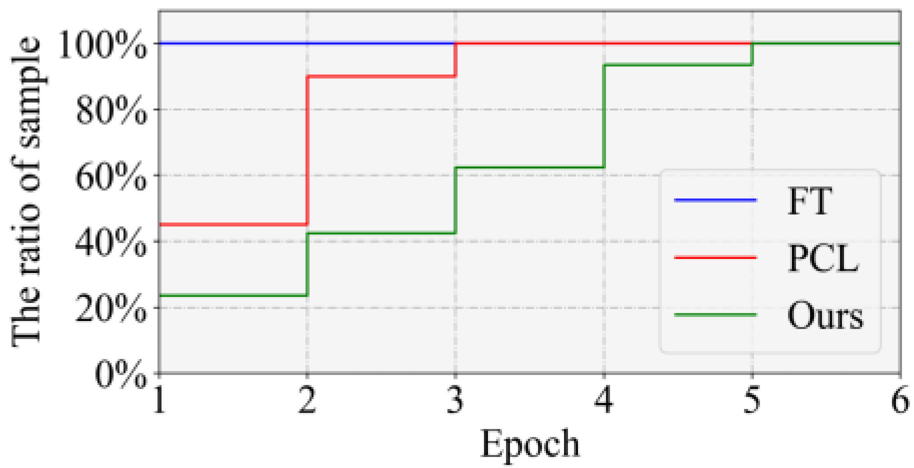


Figure 4: The ratio of training samples of different learning strategies in each epoch.

- *The training on the former subsets can be regarded as a* ***pre-training process*** *that helps model optimization and* ***regularizes the training*** *on the later subsets.*

Effectiveness of Prior Knowledge

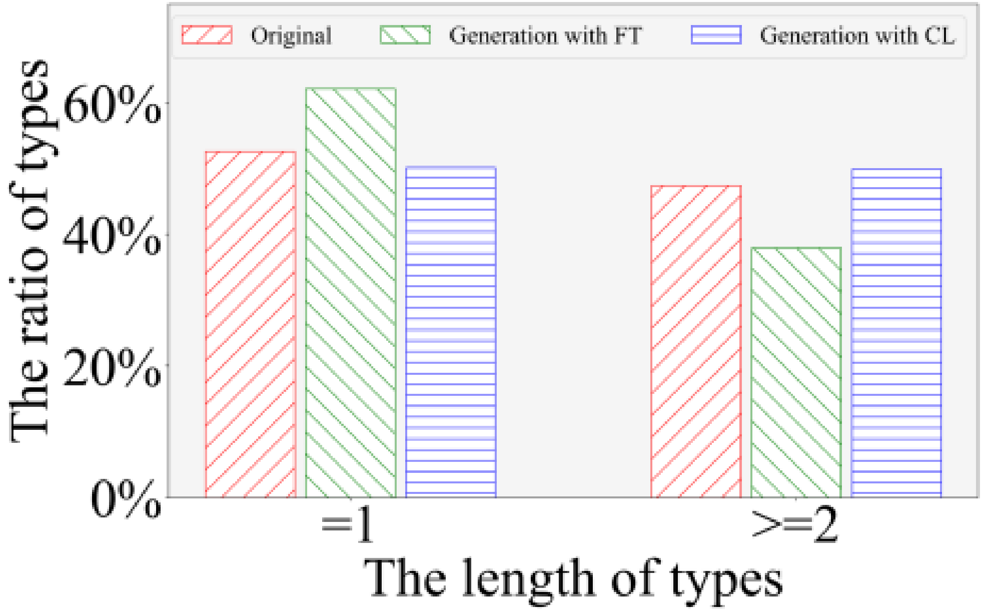
知識工場

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**

Table 6: Performance comparisons of our model and its variants on the auto-generated and human-annotated test set.

- *If the prior knowledge is ignored, SPL would only rely on the self-judgment of the model and* ***treat all the selected samples equally***

Analysis of Prior Knowledge



- *The prior knowledge about the type length is considered to **re-weight the importance of samples**, which lets the model **pay more attention to fine-grained types.***

Applications

知識工場

Short Text Classification

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%**

Table 7: Performance of short text classification based on Bi-LSTM without/with different external knowledge on NLPCC2017 dataset.

Entity linking

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

Table 8: Performance of entity linking model DCA-SL with different external knowledge.

- *The generated types effectively improve downstream tasks' performance.*

Conclusion

- Propose **a novel generative paradigm** for entity typing
- Employs a generative PLM trained with **curriculum learning**
- The prior knowledge of **type length** and **subset order** help our model generate more high-quality fine-grained types.

Thanks