Part 2

Efficient Knowledge Graph Construction

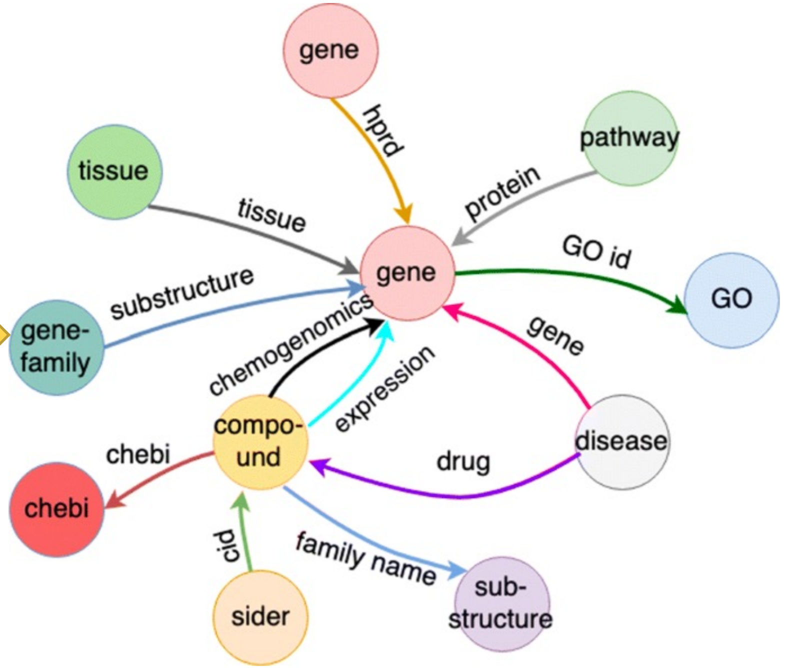
Ningyu Zhang

Knowledge Graph Construction

Information extraction for KG construction

Text2Knowledge

Language Pretrain
BERT、GPT…..



属性：字

张之洞，晚清名臣、清代洋务派代表人物，祖籍直隶南皮，出生于贵州，字孝达。

实体 属性值

isSpouseOf

Jack is married to the microbiologist known as **Dr. Germ** in the USA.

Entity Pair

Data! Money! Time!

Medical Text

今日查房，患者无发热，无头晕，无呕吐，左肩疼痛缓解，胃纳一般，大小便自解，一般情况可。查体：T36.9° C，P75次/分，R19次/分，BP129/78mmHg。头颅无畸形，心肺听诊无殊，腹平软无压痛，肝脾肋下未及，莫菲征阴性。左肩部肿胀缓解，左肩悬吊制动在位，左肩活动受限，肢端血循好。

血压 —数值→ BP129/78mmHg

心跳 —数值→ P75次/分

Annotation

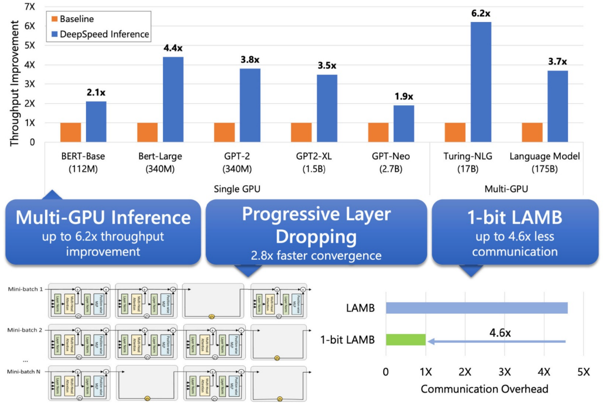
Expert Annotation Cost!

GPU Computation Cost!

Inference Time Cost!

Training

Inference



Picture from https://www.microsoft.com/en-us/research/blog/deepspeed-accelerating-large-scale-model-inference-and-training-via-system-optimizations-and-compression/

Outline

- ❑What have been done?
  - **Cutting-edge** Methods
- ❑What can we do?
  - Remaining **Issues** and **Future Works**
- ❑How can we use?
  - Open-sourced **Toolkits**

What have been done?

Cutting-edge Methods

Scope: Efficient Knowledge Graph Construction

Data Efficiency: few-shot learning with different types of data

- Prompt learning, Retrieval-augmentation

Model Efficiency: parameter-efficient learning, unified architecture

- Parameter-efficient learning

Inference Efficiency: fast Inference

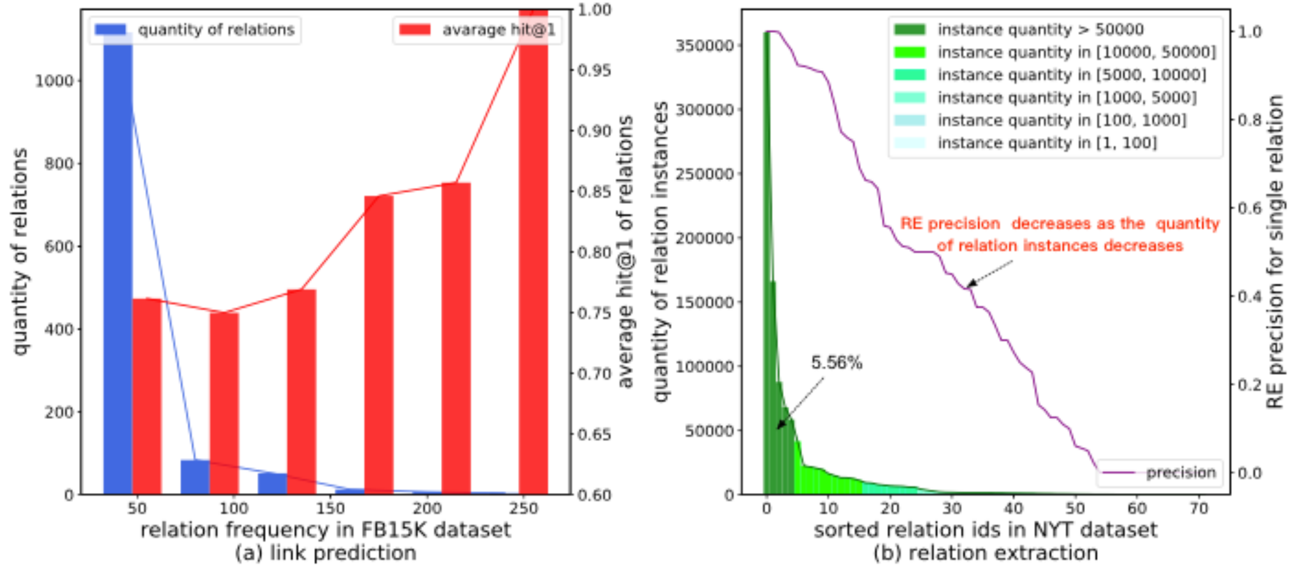
- FastRE, sequence-to-sequence

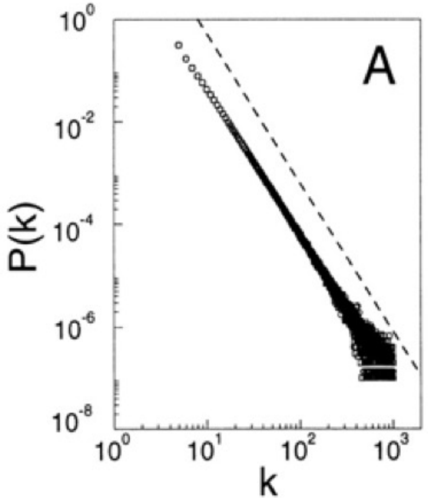
Why **Data-Efficient** Knowledge Graph Construction?

Data Sparsity **+** Annotation Cost

Knowledge: Long-tailed distribution

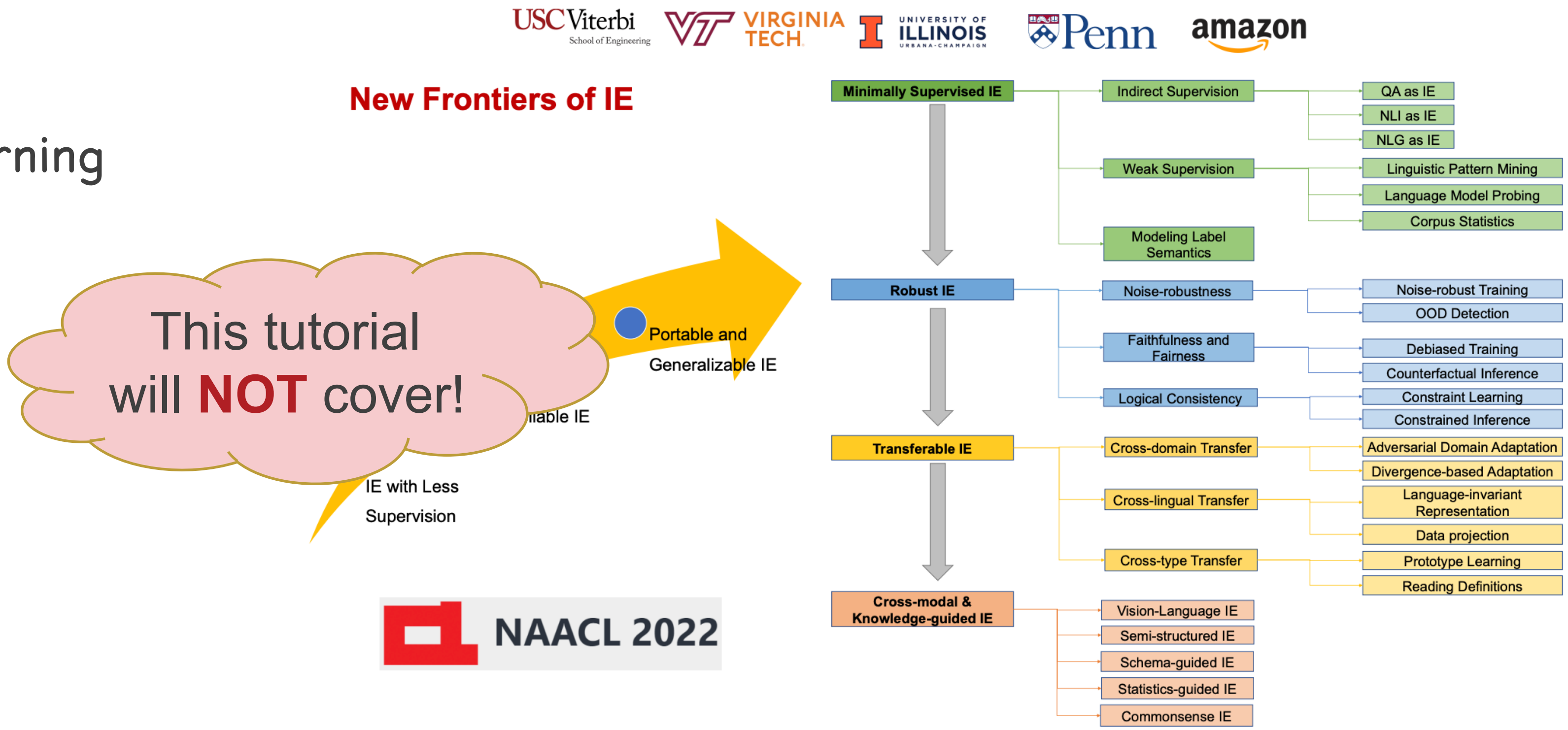
Graph: Power-law distributions



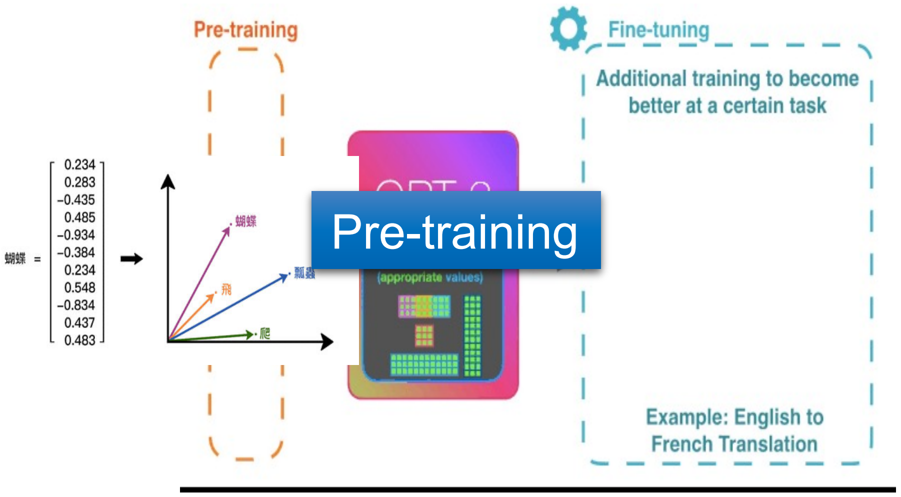


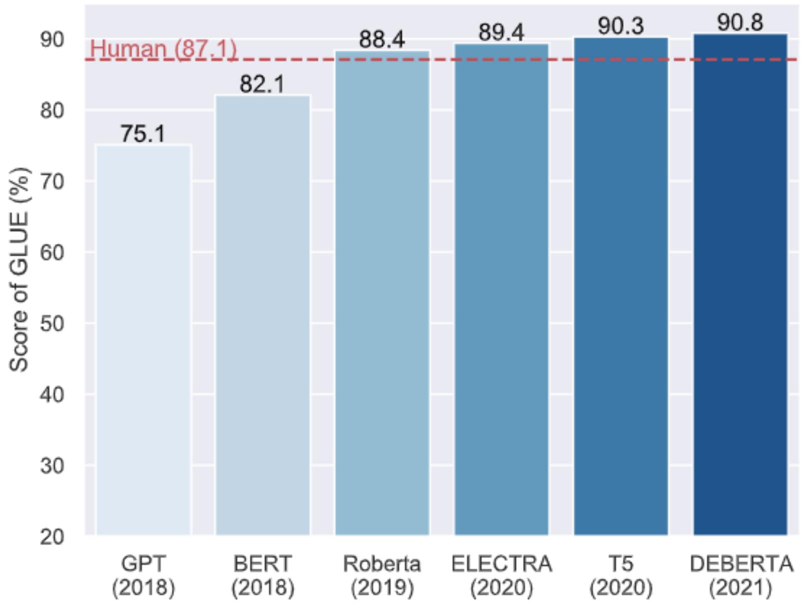
Outline

- ❑ Transfer Learning
- ❑ Minimally Supervised Learning
- ❑ Few/Zero-shot Learning
- ❑ Data Augmentation
- ❑ Semi-supervised Learning



Pre-trained Language Models





Prompt Learning for Pre-trained Language Models

Prompt learning from GPT3 : tremendous promise in few-shot/zero-shot NLP

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>
```

task description

examples

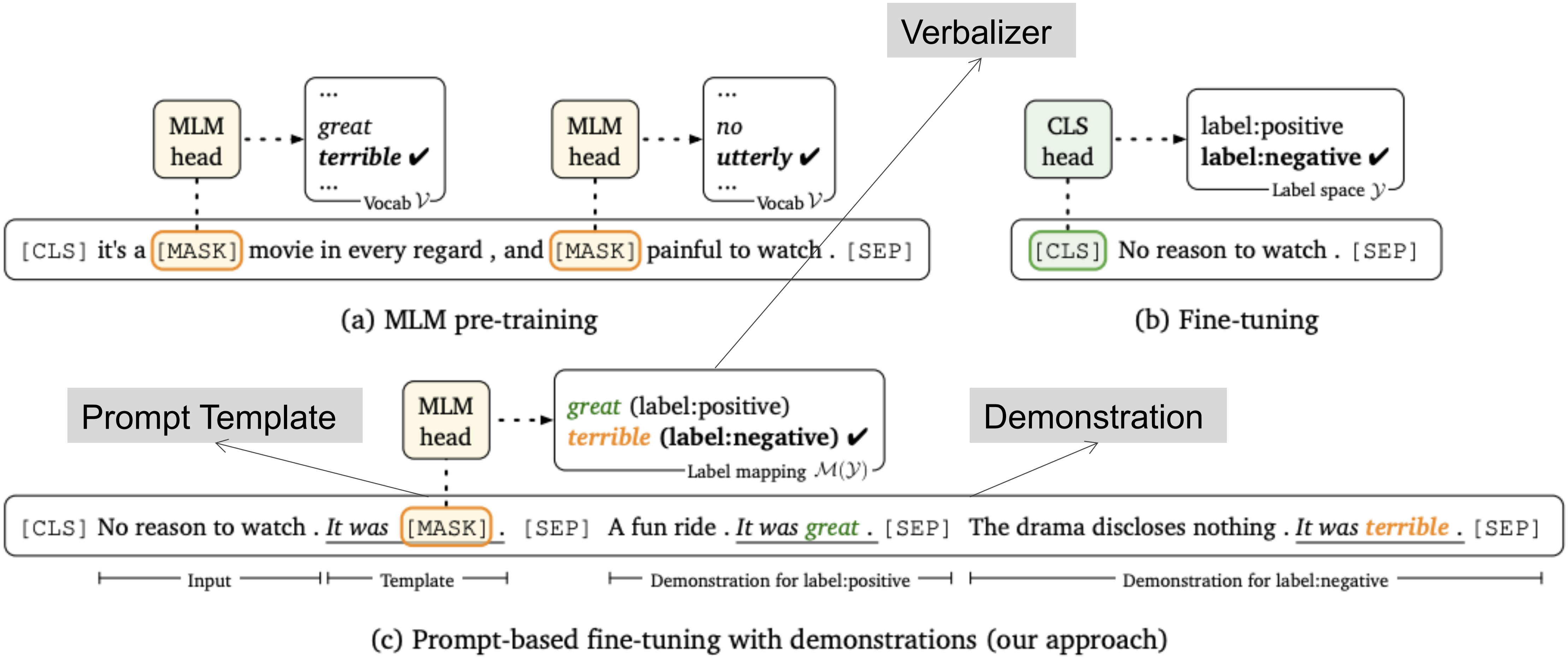
prompt

Language Models Are Few-shot Learners (NeurIPS 2020)

Prompt Learning

AACL IJCNLP 2022

Prompt learning for data-efficient NLP

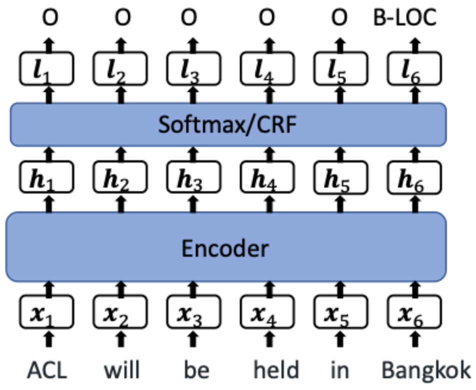


Making Pre-trained Language Models Better Few-shot Learners (ACL2021)

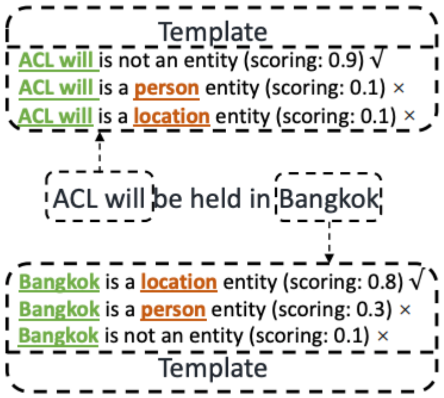
Prompt Learning

AACL IJCNLP 2022

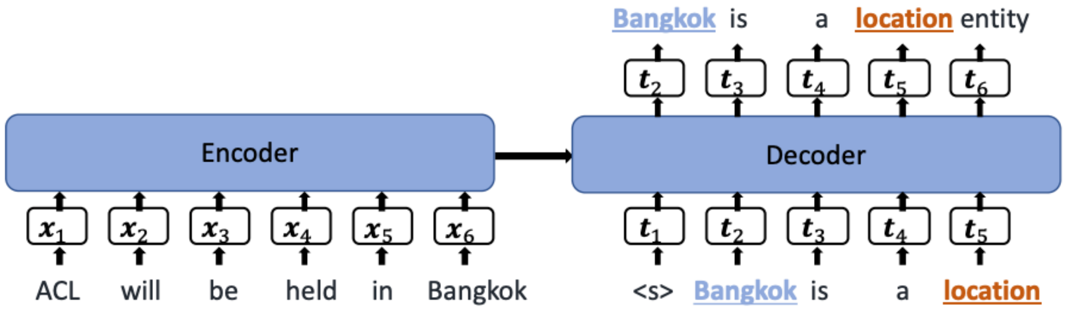
Prompt learning for data-efficient NER



(a) Traditional sequence labeling method.



(b) Inference of template-based method.



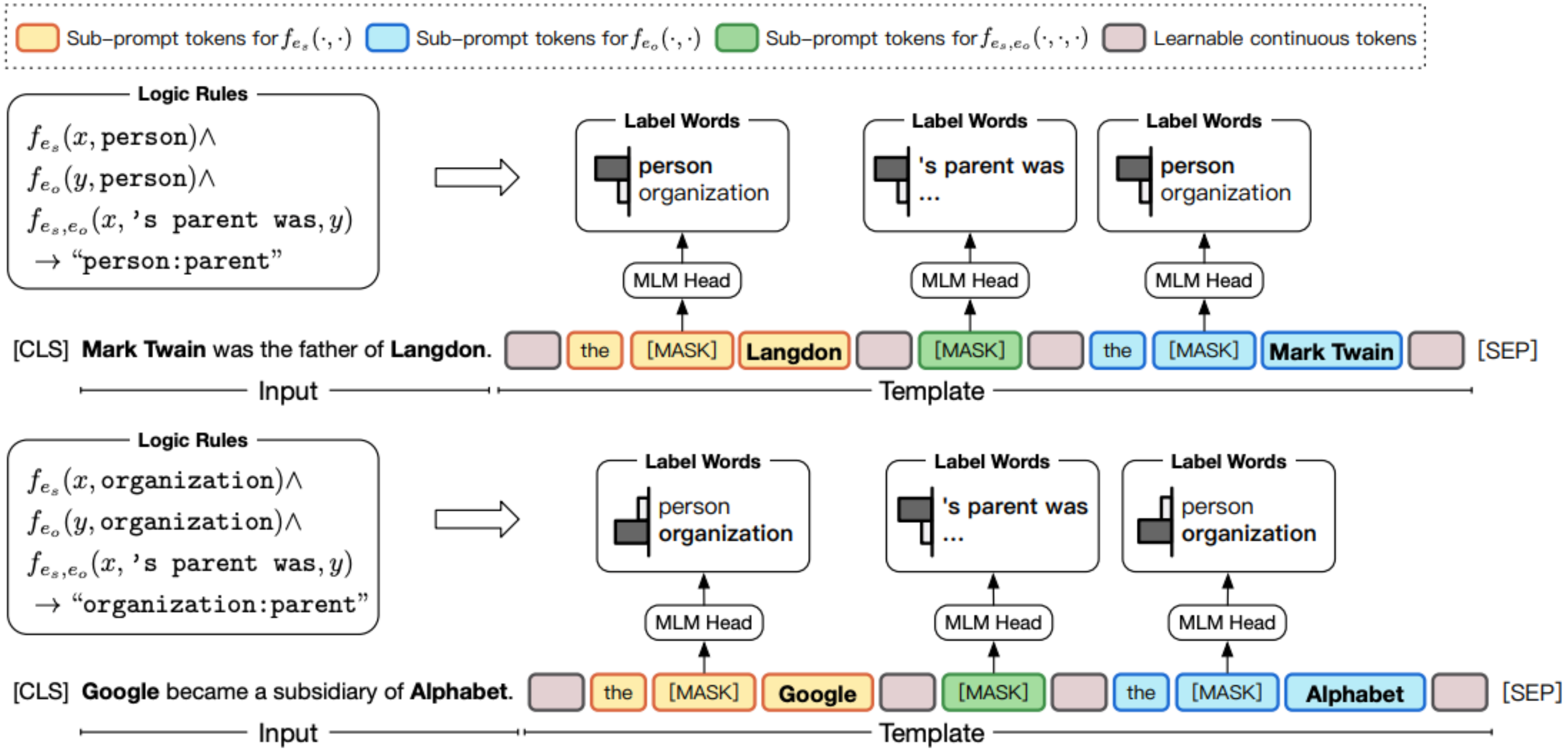
(c) Training of template-based method. The template we use here is "$\langle x_{i:j} \rangle$ is a $\langle y_k \rangle$ entity".

Source	Methods	10	20	50	100	200	500
	MIT Movie						
None	Sequence Labeling BERT	25.2	42.2	49.64	50.7	59.3	74.4
	Template-based BART	37.3	48.5	52.2	56.3	62.0	74.9
CoNLL03	Wiseman and Stratos (2019)	3.1	4.5	4.1	5.3	5.4	8.6
	Ziyadi et al. (2020)	40.1	39.5	40.2	40.0	40.0	39.5
	Huang et al. (2020)*	36.4	36.8	38.0	38.2	35.4	38.3
	Sequence Labeling BERT	28.3	45.2	50.0	52.4	60.7	76.8
	Sequence Labeling BART	13.6	30.4	47,8	49.1	55.8	66.9
	Template-based BART	42.4	54.2	59.6	65.3	69.6	80.3
	MIT Restaurant						
None	Sequence Labeling BERT	21.8	39.4	52.7	53.5	57.4	61.3
	Template-based BART	46.0	57.1	58.7	60.1	62.8	65.0
CoNLL03	Wiseman and Stratos (2019)	4.1	3.6	4.0	4.6	5.5	8.1
	Ziyadi et al. (2020)	27.6	29.5	31.2	33.7	34 .5	34.6
	Huang et al. (2020)	46.1	48.2	49.6	50.0	50.1	
	Sequence Labeling BERT	27.2	40.9	56.3	57.4	58.6	75.3
	Sequence Labeling BART	8.8	11.1	42.7	45.3	47.8	58.2
	Template-based BART	53.1	60.3	64.1	67.3	72.2	75.7
	ATIS						
None	Sequence Labeling BERT	44.1	76.7	90.7	-	-	-
	Template-based BART	71.7	79.4	92.6	-	-	-
CoNLL03	Wiseman and Stratos (2019)	6.7	8.8	11.1	-	-	-
	Ziyadi et al. (2020)	17.4	19.8	22.2	-	-	-
	Huang et al. (2020)	71.2	74.8	76.0	-	-	-
	Sequence Labeling BERT	53.9	78.5	92.2	-	-	-
	Sequence Labeling BART	51.3	74.4	89.9	-	-	-
	Template-based BART	77.3	88.9	93.5	-	-	-

Template-Based Named Entity Recognition Using BART (ACL2021)

Prompt Learning

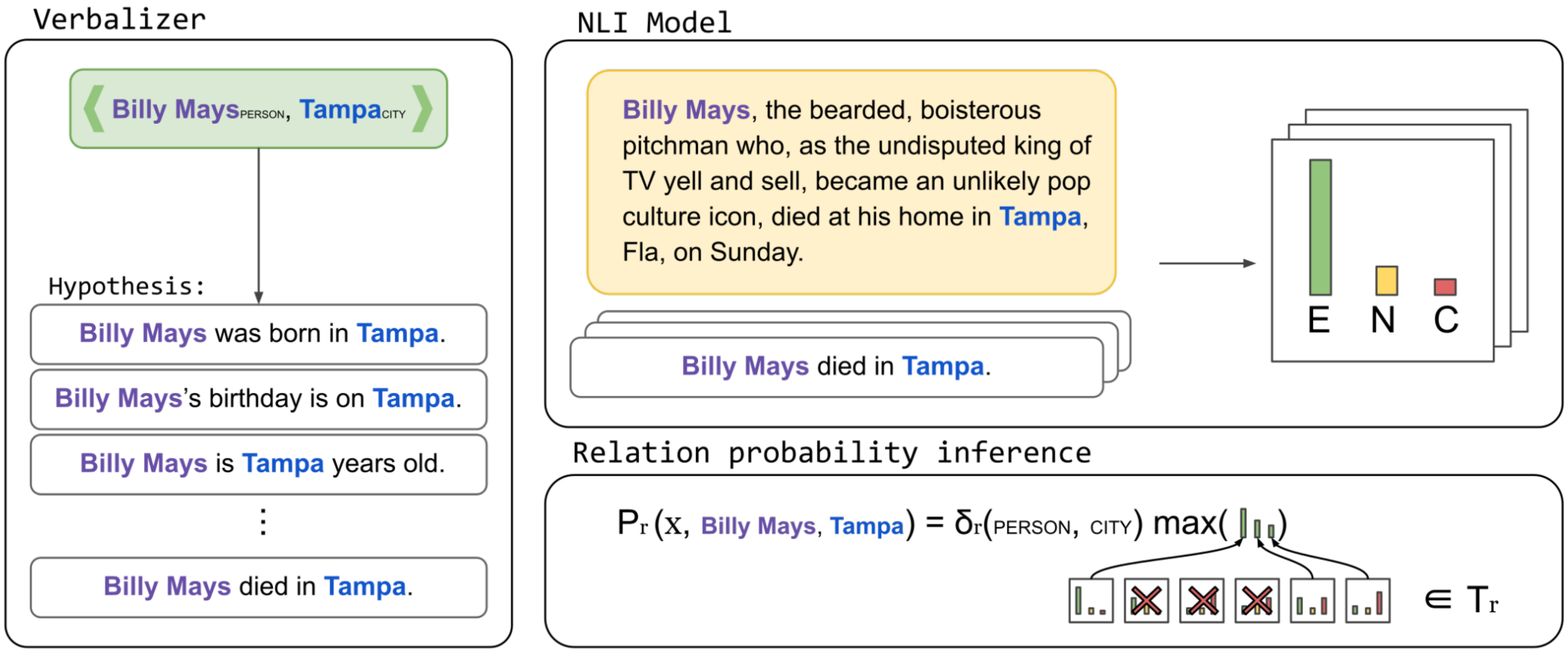
Prompt learning for data-efficient RE



PTR: Prompt Tuning with Rules for Text Classification 2021

Prompt Learning

Label Verbalization and Entailment



Label Verbalization and Entailment for Effective Zero- and Few-Shot Relation Extraction (EMNLP2021)

Prompt Learning

Prompting language models to generate structured texts

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

(a) Annotation samples of seen relations for training.

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

(b) Annotation samples of unseen relations for evaluation.

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

(c) Generated synthetic samples of unseen relations.

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

Define:

Dataset $D =$ (Sentences S, Triplets T, Labels Y)

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

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

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

RelationPrompt: Leveraging Prompts to Generate Synthetic Data for Zero-Shot Relation Triplet Extraction (ACL2021 Findings)

KnowPrompt: Motivation

Challenges

- ❑ **Prompt Design with Complex Semantics**

  The relation labels of *isSpouseOf* and *org : city_of_headquarters* cannot specify a single suitable label word in the vocabulary.

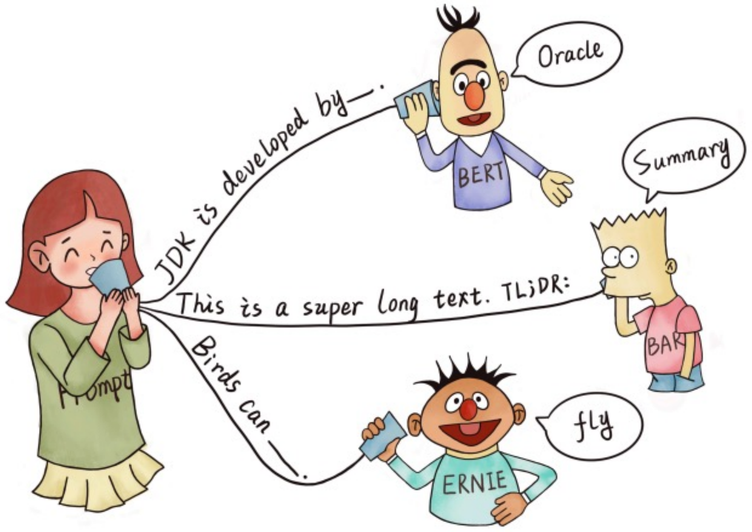
- ❑ **Inductive Bias For KG**

  If a pair of entities contains the semantics of "person" and "country", the prediction probability of the [MASK] on the relation "org:city_of_headquarters" will be lower.

isSpouseOf

Jack is married to the microbiologist known as **Dr. Germ** in the USA.

Entity Pair



Prior knowledge (Schema) in KG as a prompt to stimulate the potential knowledge in PLMs for RE

KnowPrompt: Motivation

AACL IJCNLP 2022

Challenges

- ❑ **Prompt Design with Complex Semantics**

  The relation labels of *isSpouseOf* and *org : city_of_headquarters* cannot specify a single suitable label word in the vocabulary.

- ❑ **Inductive Bias For KG**

  If a pair of entities contains the semantics of "person" and "country", the prediction probability of the [MASK] on the relation "org:city_of_headquarters" will be lower

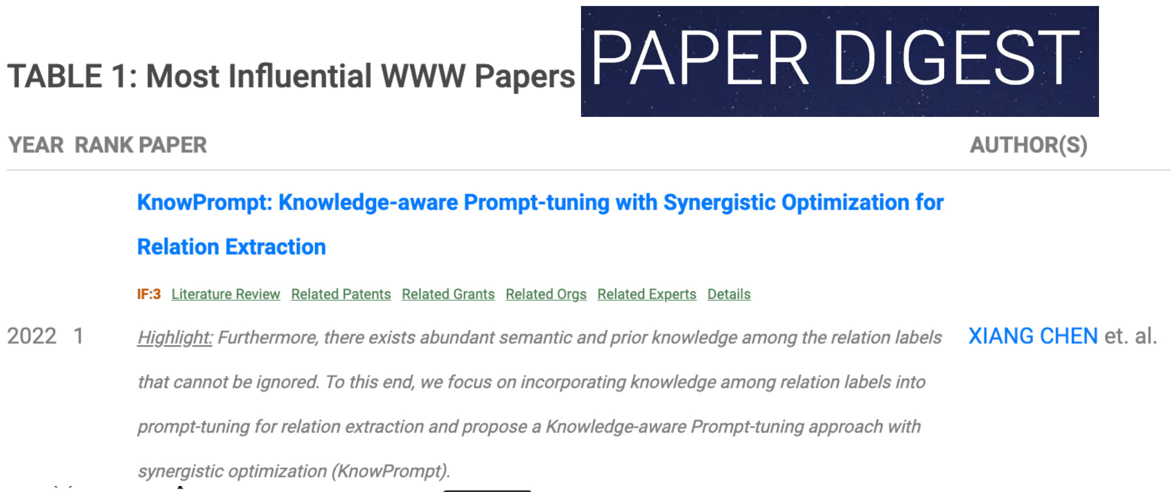
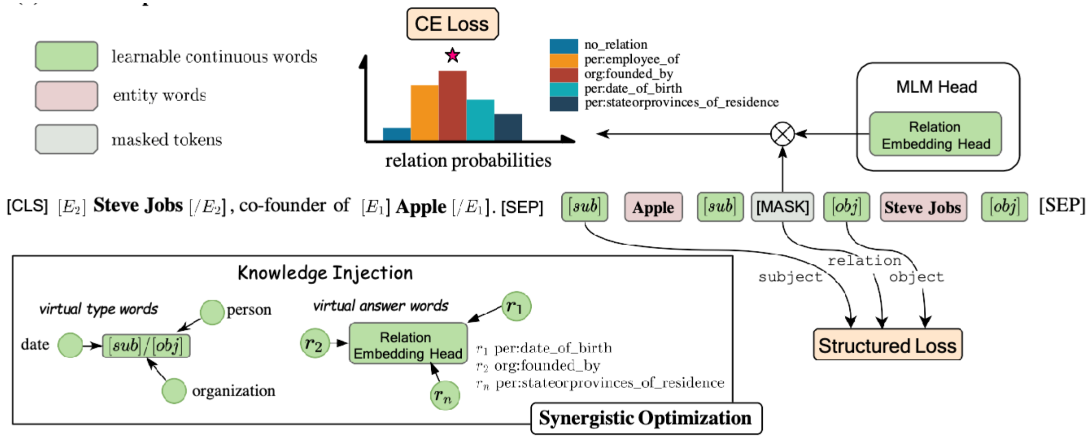


TABLE 1: Most Influential WWW Papers

PAPER DIGEST

YEAR	RANK	PAPER	AUTHOR(S)
2022	1	KnowPrompt: Knowledge-aware Prompt-tuning with Synergistic Optimization for Relation Extraction IF:3 Literature Review Related Patents Related Grants Related Orgs Related Experts Details *Highlight: Furthermore, there exists abundant semantic and prior knowledge among the relation labels that cannot be ignored. To this end, we focus on incorporating knowledge among relation labels into prompt-tuning for relation extraction and propose a Knowledge-aware Prompt-tuning approach with synergistic optimization (KnowPrompt).*	XIANG CHEN et. al.

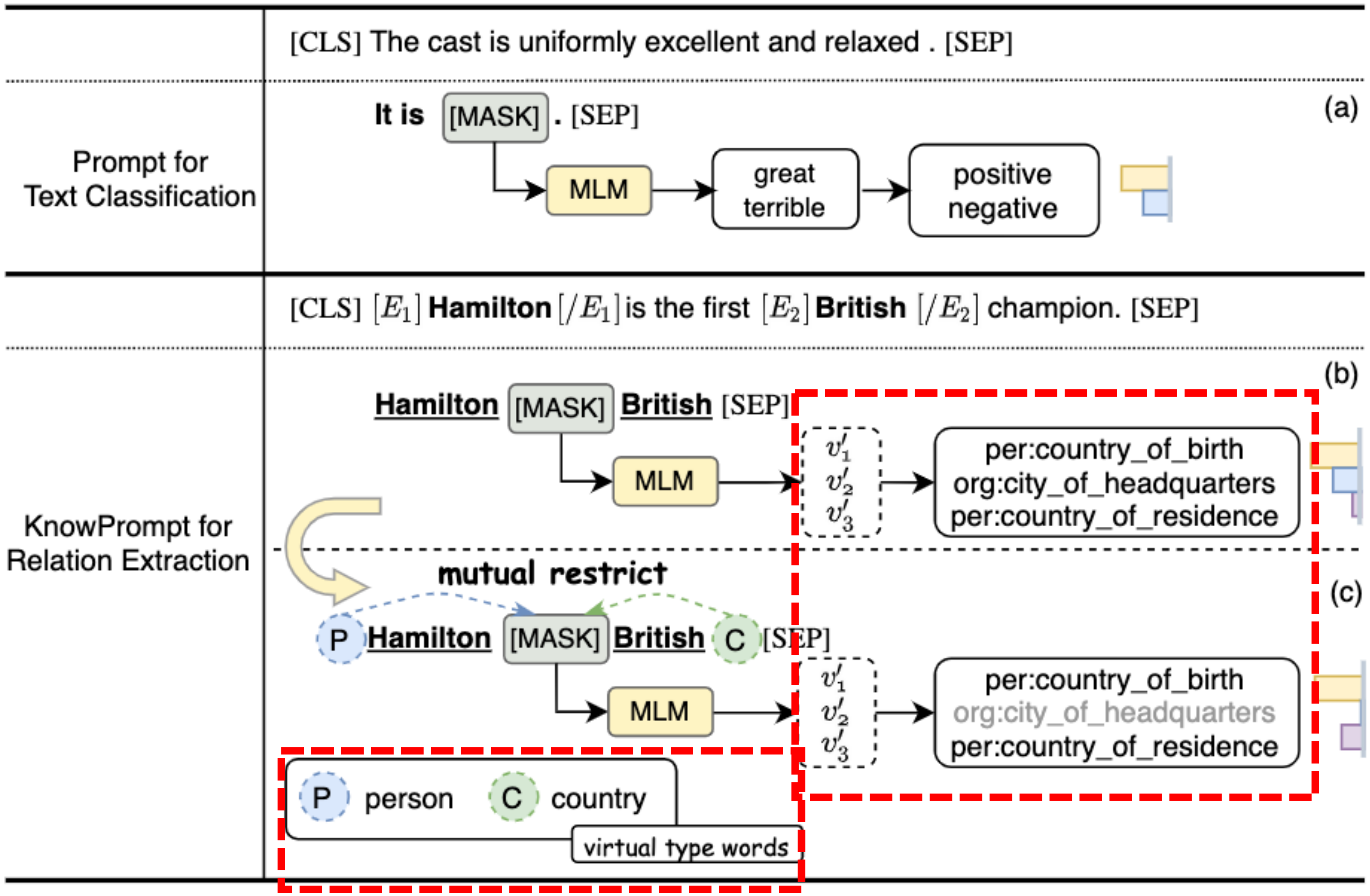


KnowPrompt'WWW22

Prior knowledge (Schema) in KG as a prompt to stimulate the potential knowledge in PLMs for RE

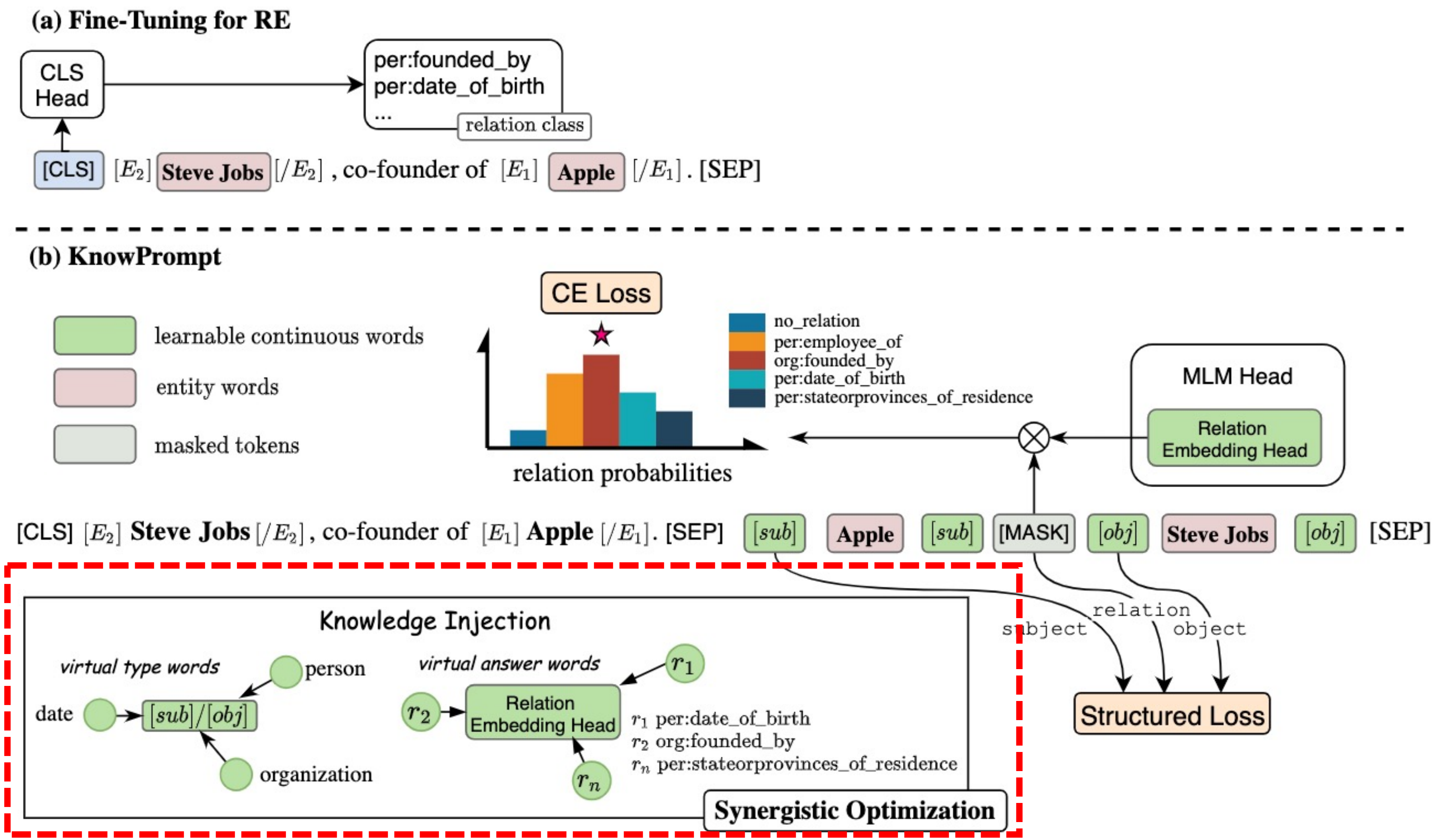
KnowPrompt: Method

From discrete to continuous: virtual embeddings as an alternative to entity and relation types



KnowPrompt: Method

Knowledge injection: constructing prior structural constraints on entity and relation



KnowPrompt: Knowledge-aware Prompt-tuning with Synergistic Optimization for Relation Extraction (WWW 2022)

KnowPrompt: Experiments

KnowPrompt can obtain better performance on **full-supervised** setting

Code is available in https://github.com/zjunlp/KnowPrompt

Standard Supervised Setting						
Methods	Extra Data	SemEval	DialogRE†	TACRED	TACRED-Revisit	Re-TACRED
Fine-tuning pre-trained models						
Fine-tuning-[RoBERTa]	w/o	87.6	57.3	68.7	76.0	84.9
SpanBERT [30]	w/	-	-	70.8	78.0	85.3
KnowBERT [38]	w/	89.1	-	71.5	79.3	89.1
LUKE [52]	w/	-	-	**72.7**	80.6	-
MTB [3]	w/	89.5	-	70.1	-	-
GDPNet [51]	w/o	-	64.9	71.5	79.3	-
Dual [2]	w/o	-	67.3	-	-	-
Prompt-tuning pre-trained models						
PTR-[RoBERTa] [22]	w/o	89.9	63.2	72.4	81.4	90.9
KnowPrompt-[RoBERTa]	w/o	**90.2** (+0.3)	**68.6** (+5.4)	72.4 (-0.3)	**82.4** (+1.0)	**91.3** (+0.4)

KnowPrompt: Experiments

AACL IJCNLP 2022

KnowPrompt can achieve large improvement on **low-resource** setting

Code is available in https://github.com/zjunlp/KnowPrompt

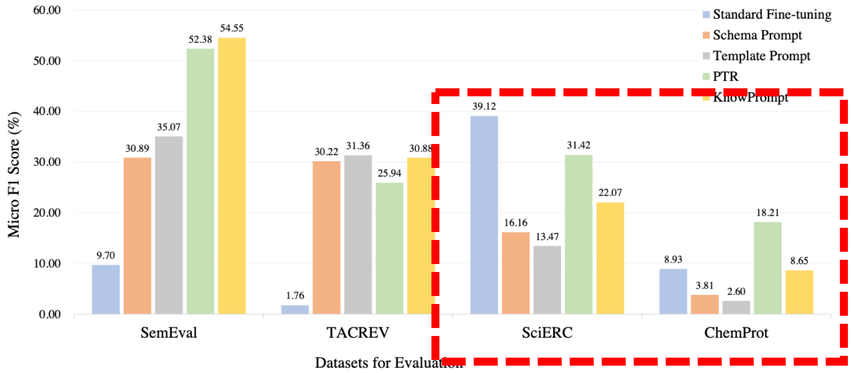
Low-Resource Setting							
Split	Methods	SemEval	DialogRE†	TACRED	TACRED-Revisit	Re-TACRED	Average
K=8	FINE-TUNING	41.3	29.8	12.2	13.5	28.5	25.1
	GDPNET	42.0	28.6	11.8	12.3	29.0	24.7
	PTR	70.5	35.5	28.1	28.7	51.5	42.9
	KNOWPROMPT	**74.3** (+33.0)	**43.8** (+14.0)	**32.0** (+19.8)	**32.1** (+18.6)	**55.3** (+26.8)	**47.5** (+22.4)
K=16	FINE-TUNING	65.2	40.8	21.5	22.3	49.5	39.9
	GDPNET	67.5	42.5	22.5	23.8	50.0	41.3
	PTR	81.3	43.5	30.7	31.4	56.2	48.6
	KNOWPROMPT	**82.9** (+17.7)	**50.8** (+10.0)	**35.4** (+13.9)	**33.1** (+10.8)	**63.3** (+13.8)	**53.1** (+13.2)
K=32	FINE-TUNING	80.1	49.7	28.0	28.2	56.0	48.4
	GDPNET	81.2	50.2	28.8	29.1	56.5	49.2
	PTR	84.2	49.5	32.1	32.4	62.1	52.1
	KNOWPROMPT	**84.8** (+4.7)	**55.3** (+3.6)	**36.5** (+8.5)	**34.7** (+6.5)	**65.0** (+9.0)	**55.3** (+6.9)

KnowPrompt: Knowledge-aware Prompt-tuning with Synergistic Optimization for Relation Extraction (WWW 2022)

KnowPrompt: More results on LREBench

KnowPrompt obtain better performance on **most** datasets

Code is available in https://github.com/zjunlp/LREBench



❑ Finding 1

Prompt-based tuning largely outperforms standard fine-tuning

❑ Finding 3

Balancing methods may fail on those challenging datasets

Dataset	Metric	Fine-Tune Normal 8-shot	Fine-Tune Normal 10%	Fine-Tune Normal 100%	Fine-Tune Balance 10%	Fine-Tune Balance 100%	Fine-Tune DA 10%	Fine-Tune ST 10%	Prompt Normal 8-shot	Prompt Normal 10%	Prompt Normal 100%	Prompt Balance 10%	Prompt Balance 100%	Prompt DA 10%	Prompt ST 10%
SemEval	MaF1	2.69	34.63	81.88	41.84	82.44	69.84	60.10	58.89	44.71	83.40	54.54	83.20	71.73	63.55
	MiF1	9.70	54.61	89.10	58.26	89.44	78.98	74.12	62.52	69.90	90.01	76.53	92.31	83.54	76.81
TACREV	MaF1	1.02	47.32	63.41	48.64	63.38	50.68	48.84	29.46	61.40	67.08	63.09	69.63	62.20	7.32
	MiF1	1.76	65.43	71.68	67.19	73.86	65.99	66.89	30.88	77.00	78.30	76.25	81.41	76.90	32.93
Wiki80	MaF1	37.89	37.82	71.31	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	63.99	83.72	63.40	60.86
	MiF1	44.85	46.50	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	67.86	83.86	66.96	65.04
SciERC	MaF1	10.41	10.31	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	84.83	65.98	56.94
	MiF1	39.12	54.66	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	76.90	90.04	79.92	76.32
ChemProt	MaF1	2.18	27.96	47.35	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	36.43	47.16	38.99	47.07	37.44	33.62
	MiF1	8.93	49.20	68.81	54.98	68.77	56.58	54.17	8.65	56.96	69.14	57.28	69.12	58.26	53.55
DialogRE	MaF1	1.13	2.17	25.31	5.84	27.28	-	0.00	44.96	45.42	69.99	49.68	69.73	-	45.43
	MiF1	3.92	23.37	41.52	24.53	41.24	-	0.00	45.70	58.66	76.20	58.16	75.88	-	57.94
DuIE2.0	MaF1	36.62	90.46	95.01	92.91	96.00	-	89.27	80.31	93.48	95.73	93.70	96.01	-	90.49
	MiF1	39.00	94.42	96.22	94.46	96.13	-	93.81	82.14	95.09	96.43	95.23	96.44	-	93.35
CMeIE	MaF1	13.68	62.30	84.37	67.22	86.31	-	58.46	36.54	67.59	86.42	67.84	86.68	-	65.79
	MiF1	17.05	79.82	90.48	80.43	90.56	-	78.92	38.02	83.38	92.08	83.40	92.14	-	81.26

Eight datasets of single-sentence & document with various languages

❑ Finding 2

Data augmentation achieves much gain on RE

❑ Finding 4

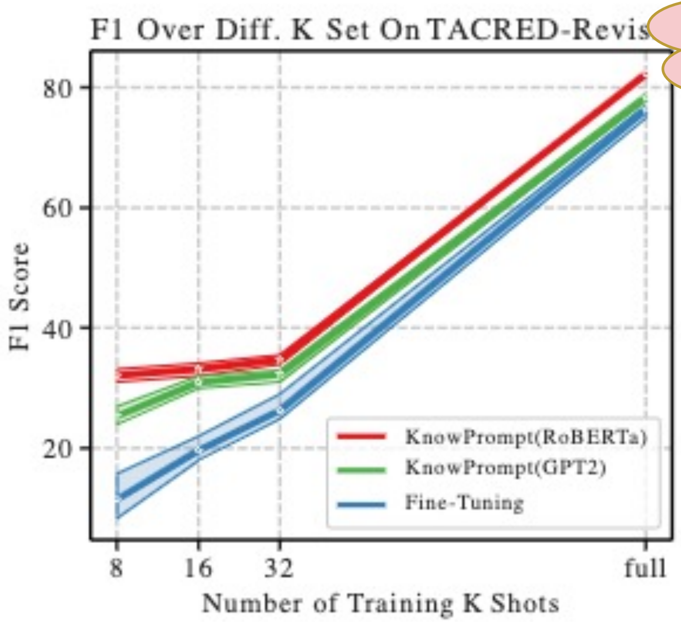
RE systems may fail on cross-sentence contexts and among multiple triples

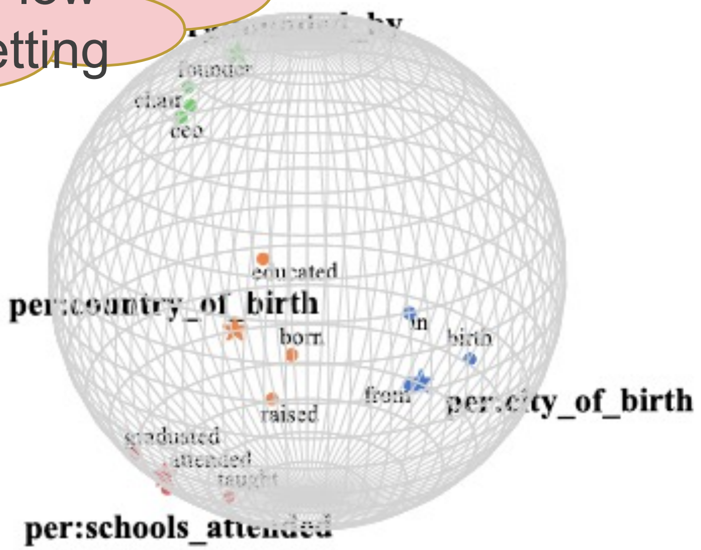
Towards Realistic Low-resource Relation Extraction: A Benchmark with Empirical Baseline Study (EMNLP2022 Findings)

KnowPrompt: Analysis

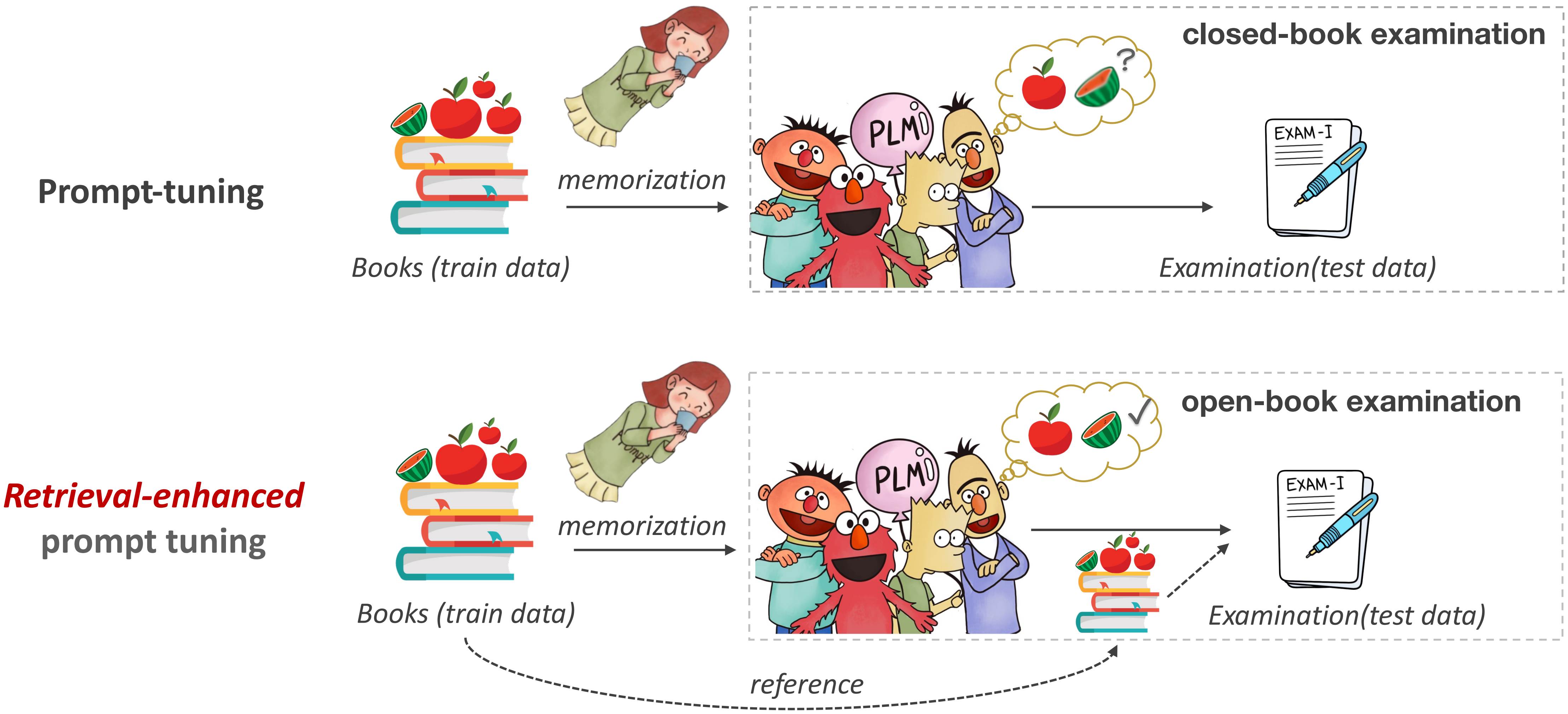
Input Example of our KnowPrompt	Top 3 words around [sub]	Top 3 words around [obj]
x:[CLS] It sold [E_1] **ALICO** [/E_1] to [E_2] **MetLife Inc** [E_2] for \$ 162 billion. [SEP] [sub] **ALICO** [sub] [MASK] [obj] **MetLife Inc** [obj]. [SEP] **y**: "*org : member_of*"	organization group corporation	company plc organization
x: [CLS] [E_1] **Ismael Rukwago** [/E_1], a senior [E_2] **ADF** [E_2] commander, denied any involvement. [SEP] [sub]**Ismael Rukwago** [sub] [MASK] [obj] **ADF** [obj]. [SEP] **y**: "*per : employee_of*"	person commander colonel	intelligence organization command

Knowprompt fail on **extreme** low-resource setting



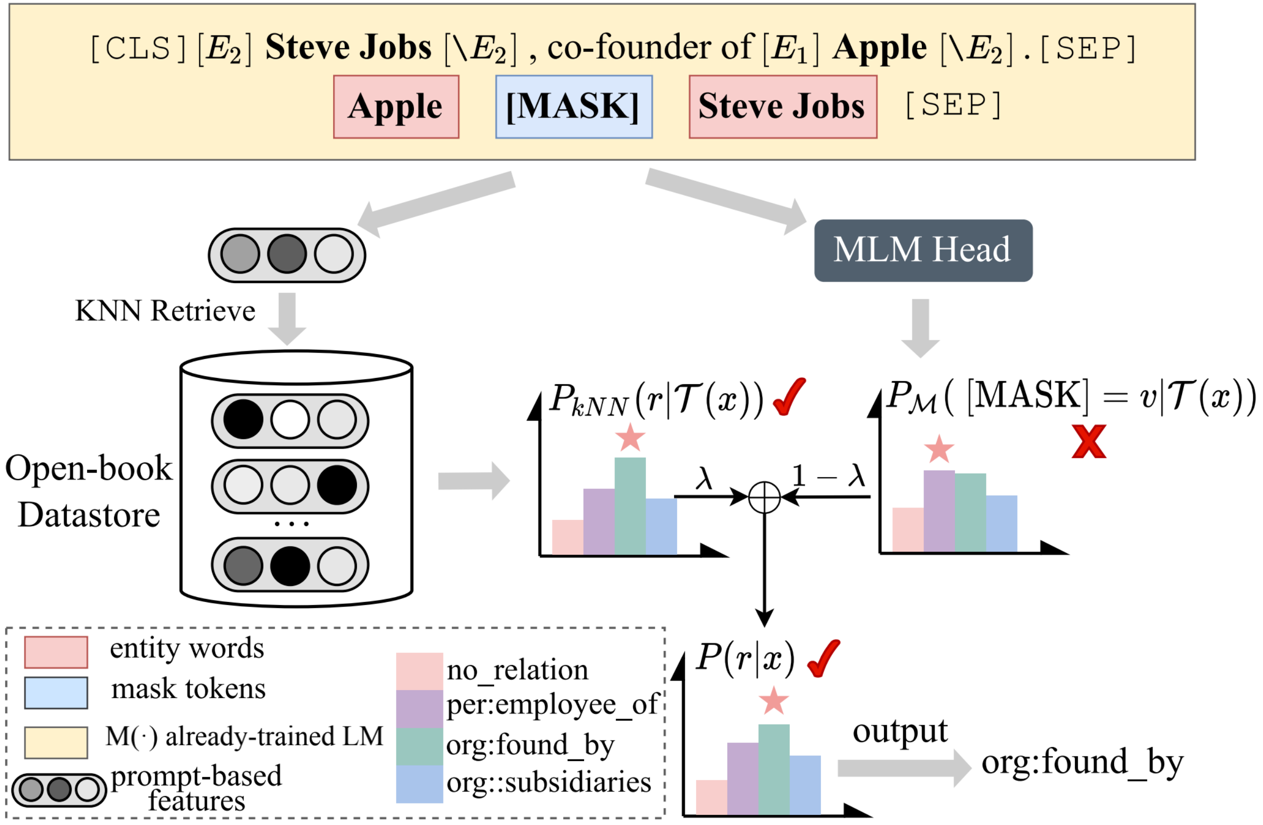


What's Missing? More knowledge in Training Data



Relation Extraction as Open-book Examination: Retrieval-enhanced Prompt Tuning (SIGIR2022 short)

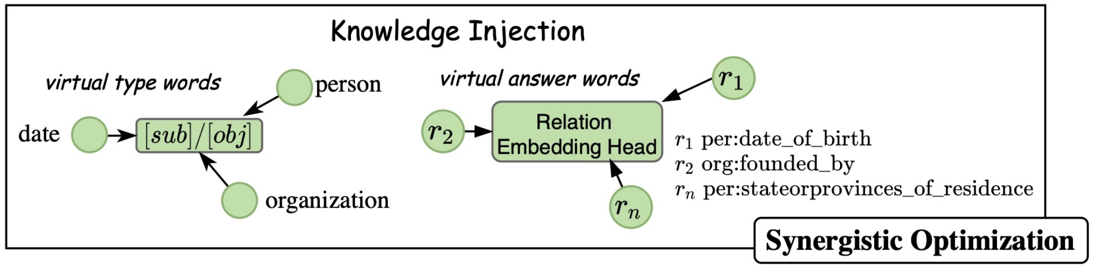
RetrievalRE



$$P_{\mathcal{M}}(\text{[MASK]} = v|\mathcal{T}(x))$$

$$P_{k\text{NN}}(r \mid \mathcal{T}(x)) \propto \sum_{(c_i,r_i)\in\mathcal{N}} 1_{r=r_i} \exp\left(-d\left(h_x, h_{c_i}\right)\right).$$

$$P(r \mid x) = \lambda P_{k\text{NN}}(r \mid \mathcal{T}(x)) + (1-\lambda) P_{\mathcal{M}}(\text{[MASK]} = v|\mathcal{T}(x)).$$



*Soft label words & Knowledge Injection (from **KnowPrompt**[1])*

Experiments

Code is available in https://github.com/zjunlp/PromptKG/tree/main/research/RetrievalRE

Table 1: Model performance of RE models in the low-resource setting. We report the mean and standard deviation performance of micro F_1 scores (%) over 5 different splits. The best numbers are highlighted in each column.

Source	Model	SemEval			TACRED			TACREV		
		K=1	K=5	K=16	K=1	K=5	K=16	K=1	K=5	K=16
None	FINE-TUNING	18.5 (± 1.4)	41.5 (± 2.3)	66.1 (± 0.4)	7.6 (± 3.0)	16.6 (± 2.1)	26.8 (± 1.8)	7.2 (± 1.4)	16.3 (± 2.1)	25.8 (± 1.2)
	GDPNET	10.3 (± 2.5)	42.7 (± 2.0)	67.5 (± 0.8)	4.2 (± 3.8)	15.5 (± 2.3)	28.0 (± 1.8)	5.1 (± 2.4)	17.8 (± 2.4)	26.4 (± 1.2)
	PTR	14.7 (± 1.1)	53.9 (± 1.9)	80.6 (± 1.2)	8.6 (± 2.5)	24.9 (± 3.1)	30.7 (± 2.0)	9.4 (± 0.7)	26.9 (± 1.5)	31.4 (± 0.3)
	KnowPrompt	28.6 (± 6.2)	66.1 (± 8.6)	80.9 (± 1.6)	17.6 (± 1.8)	28.8 (± 2.0)	34.7 (± 1.8)	17.8 (± 2.2)	30.4 (± 0.5)	33.2 (± 1.4)
	RetrievalRE	**33.3** (± 1.6)	**69.7** (± 1.7)	**81.8** (± 1.0)	**19.5** (± 1.5)	**30.7** (± 1.7)	**36.1** (± 1.2)	**18.7** (± 1.8)	**30.6** (± 0.2)	**35.3** (± 0.3)

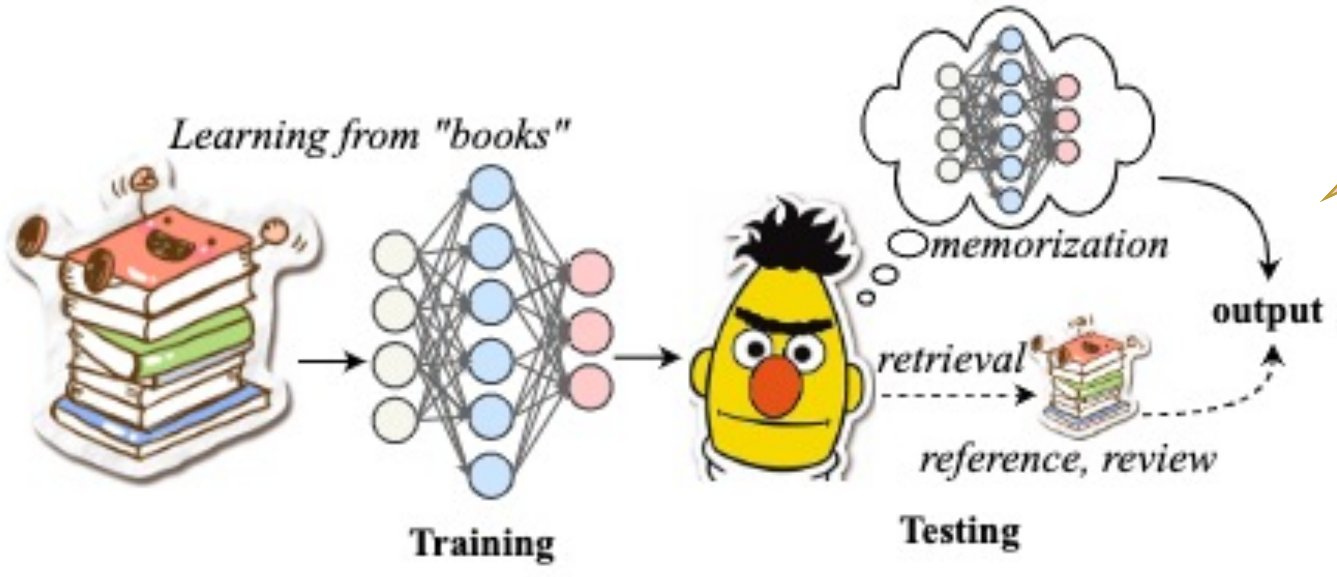
Table 3: Standard RE performance of F_1 scores (%) on different test sets. Best results are bold.

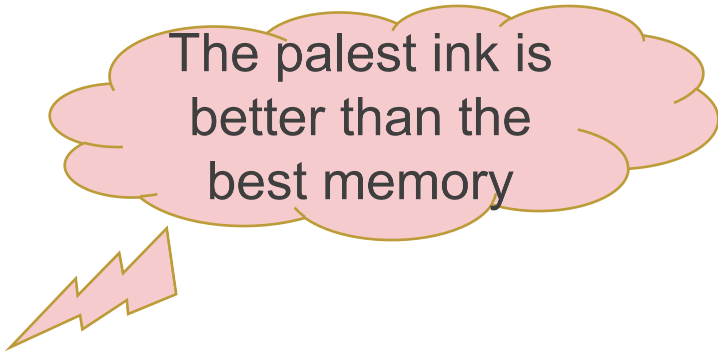
Standard Supervised Setting					
Methods	SemEval	DialogRE	TACRED	TACREV	Re-TACRED
Fine-tuning pre-trained models					
FINE-TUNING	87.6	57.3	68.7	76.0	84.9
SPANBERT [22]	-	-	70.8	78.0	85.3
KNOWBERT [30]	89.1	-	71.5	79.3	89.1
LUKE [38]	-	-	**72.7**	80.6	-
MTB [4]	89.5	-	70.1	-	-
GDPNET [37]	-	64.9	71.5	79.3	-
DUAL [3]	-	67.3	-	-	-
Prompt-tuning pre-trained models					
PTR [13]	89.9	63.2	72.4	81.4	90.9
KNOWPROMPT	90.2	68.6	72.4	82.4	91.3
RETRIEVALRE	**90.4**	**69.4**	**72.7**	**82.7**	**91.5**

- Our RetrievalRE achieves improvements over all baselines in the **low-resource setting**.
- RetrievalRE can also achieve improvements of F1 scores when the training data is **exceptionally scarce**.
- Slight improvement in the **standard supervised setting**.

More Thinking

Memorisation versus **Generalisation**



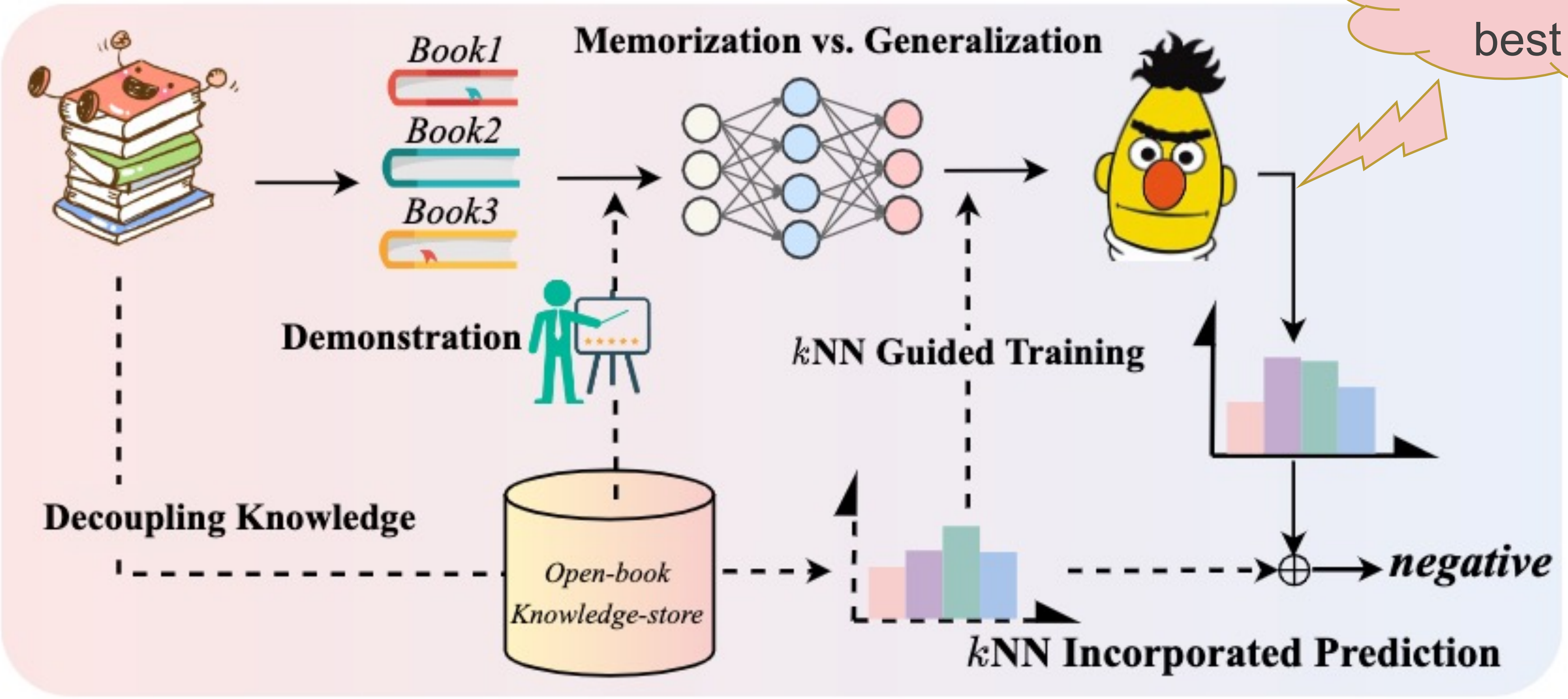


Vitaly Feldman. 2020. Does learning require memorization? a short tale about a long tail
An Empirical Study of Memorization in NLP (ACL2022)
Memorisation versus Generalisation in Pre-trained Language Models (ACL2022)

Motivation: DO NOT LEARN BY ROT !

Improve the generalization ability of prompt learning with retrieval and association.

The palest ink is better than the best memory



RetroPrompt

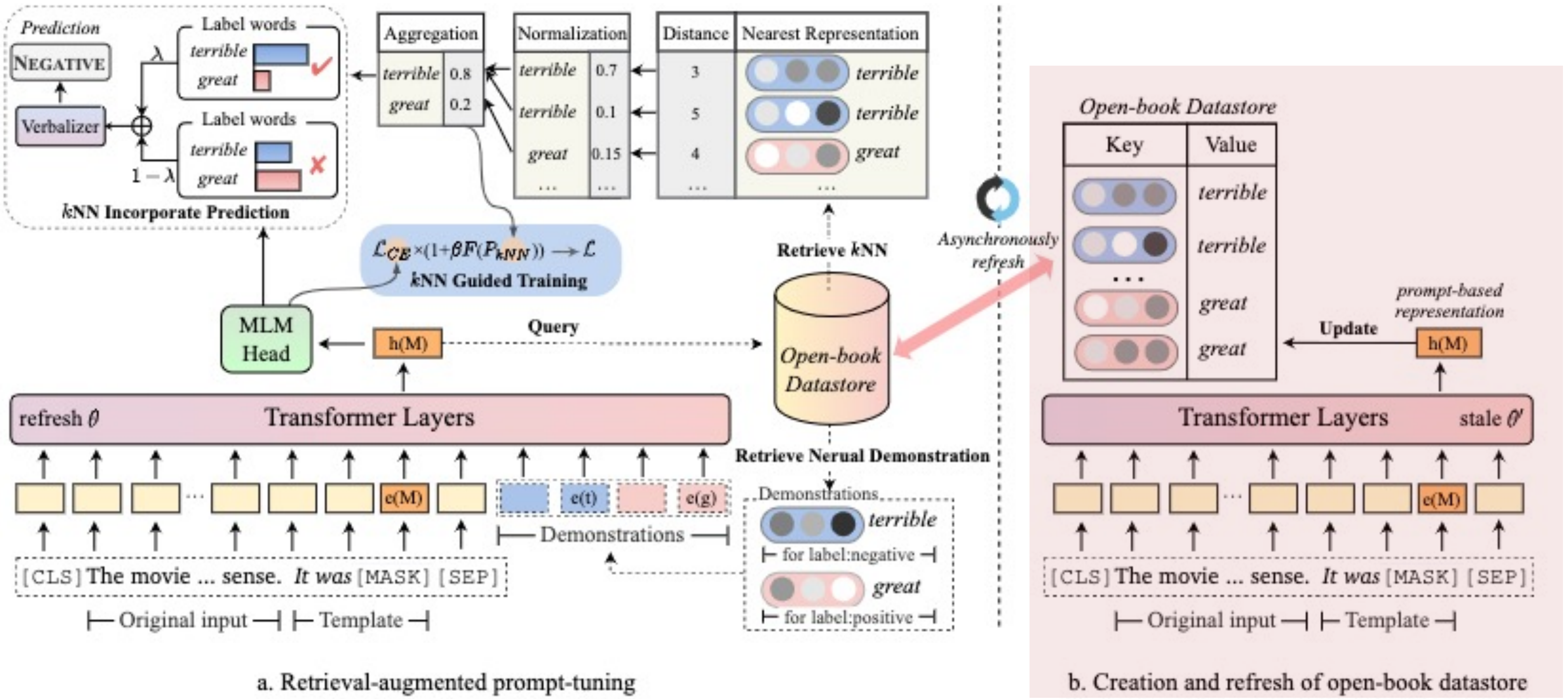


Figure 2: Overview of our RETROPROMPT. Note that $e(\cdot)$ donates the word embedding in the PLM $\mathcal{M}$, while "M","t" and "g" in the $e(\cdot)$ specifically refers to the "[MASK]", "terrible" and "great".

RetroPrompt

AACL IJCNLP 2022

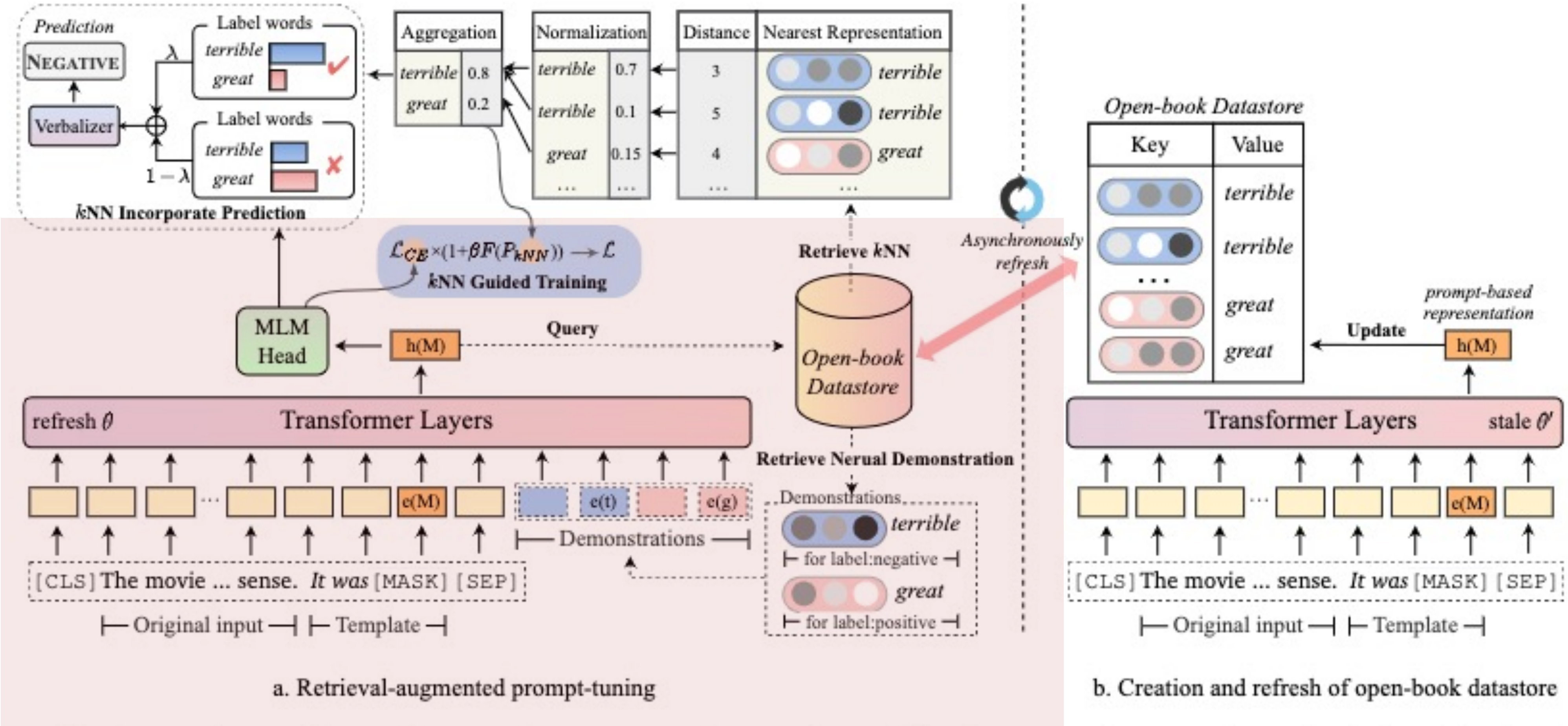
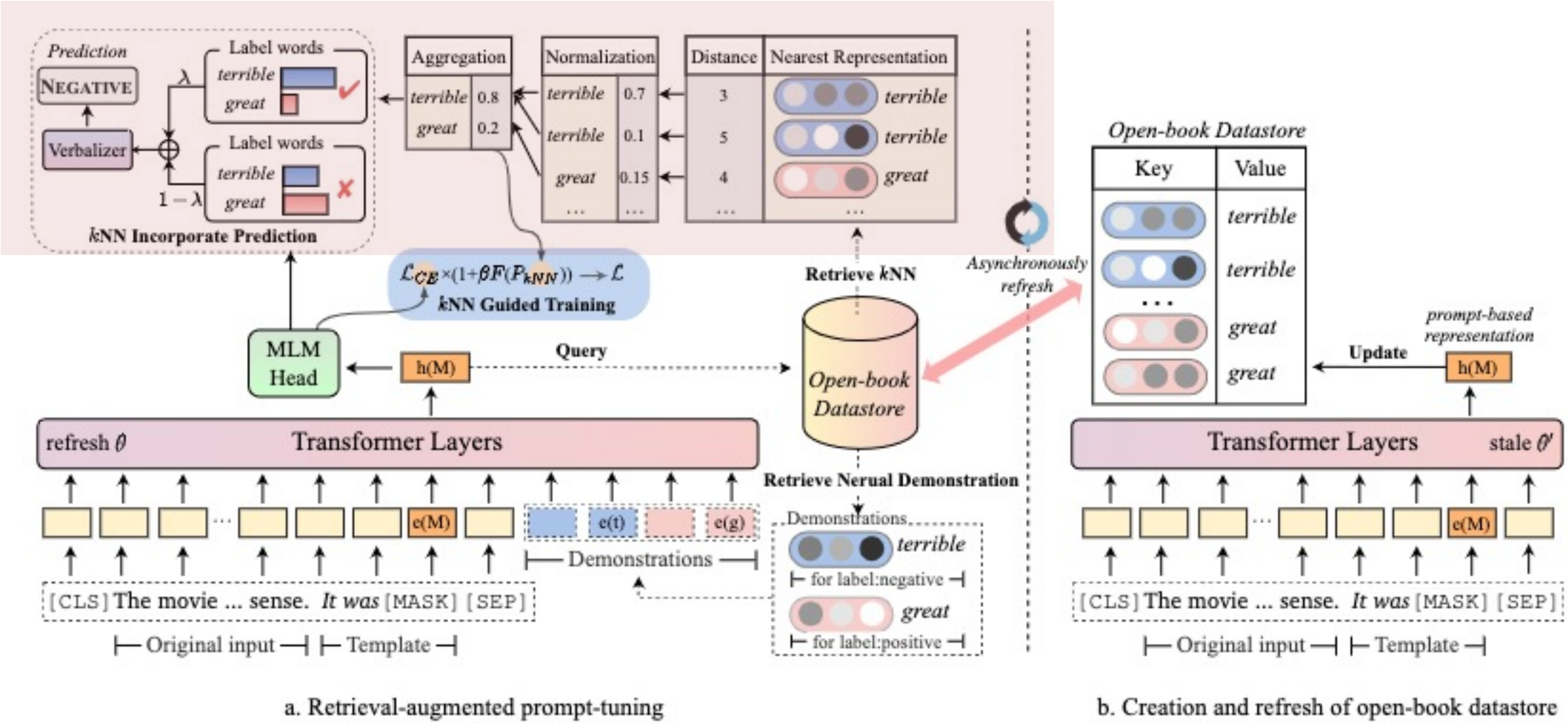


Figure 2: Overview of our RETROPROMPT. Note that $e(\cdot)$ donates the word embedding in the PLM $\mathcal{M}$, while "M","t" and "g" in the $e(\cdot)$ specifically refers to the "[MASK]", "terrible" and "great".

RetroPrompt



a. Retrieval-augmented prompt-tuning

b. Creation and refresh of open-book datastore

Figure 2: Overview of our RETROPROMPT. Note that $e(\cdot)$ donates the word embedding in the PLM $\mathcal{M}$, while “M”,“t” and “g” in the $e(\cdot)$ specifically refers to the “[MASK]”, “terrible” and “great”.

RetroPrompt

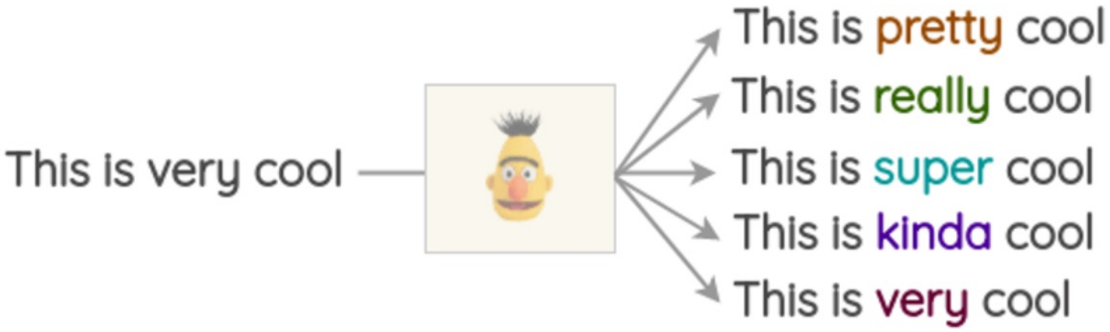
Code is available in https://github.com/zjunlp/PromptKG/tree/main/research/RetroPrompt

St.	Model	Single Sentence			Sentence Pair			Model	Information Extraction			Avg.
		SST-2 (acc)	MR (acc)	CR (acc)	MNLI (acc)	QNLI (acc)	QQP (F1)		FewN (acc)	SemEval (acc)	TACRED (F1)	
16	FT	81.4 (3.8)	76.9 (5.9)	75.8 (3.2)	45.8 (6.4)	60.2 (6.5)	60.7 (4.3)	FT	52.7 (2.2)	66.1 (1.2)	25.8 (2.8)	60.6
	LM-BFF (man)	91.6 (1.2)	87.0 (2.0)	90.3 (1.6)	64.3 (2.5)	64.6 (5.4)	65.4 (5.3)	KnPr	65.3 (1.1)	80.9 (2.5)	33.2 (2.0)	71.4
	LM-BFF (D-demo)	91.8 (1.2)	86.6 (1.8)	90.2 (1.4)	64.8 (2.3)	69.2 (5.4)	68.2 (3.2)	KnPr (D-demo)	—	—	—	72.2*
	KPT †	90.3 (1.6)	86.8 (1.8)	88.8 (3.7)	61.4 (2.1)	61.5 (2.8)	71.6 (2.7)	KPT †	65.9 (1.5)	78.8 (2.1)	32.8 (1.7)	70.9
	Ours	**93.9** (0.4)	**88.0** (0.8)	**91.9** (0.7)	**71.1** (1.8)	**71.6** (1.8)	**74.0** (2.0)	**Ours**	**67.3** (0.9)	**81.5** (1.3)	**40.7** (0.7)	**75.6**
4	FT	60.2 (2.8)	57.6 (1.4)	66.4 (5.5)	35.0 (0.3)	54.2 (3.9)	52.8 (4.7)	FT	32.7 (2.9)	38.8 (2.0)	14.7 (2.8)	45.8
	LM-BFF (man)	90.7 (0.8)	85.2 (2.8)	89.9 (1.8)	51.0 (2.5)	61.1 (6.1)	48.0 (4.9)	KnPr	52.5 (1.5)	58.4 (3.7)	28.8 (2.5)	62.8
	LM-BFF (D-demo)	90.2 (1.5)	85.5 (2.1)	89.7 (0.6)	56.1 (1.0)	61.7 (7.6)	63.2 (5.6)	KnPr (D-demo)	—	—	—	65.1*
	KPT †	88.2 (5.7)	83.4 (1.5)	87.2 (2.5)	53.7 (2.7)	59.2 (2.8)	54.9 (7.9)	KPT †	58.8 (2.2)	57.2 (3.2)	27.5 (2.2)	63.3
	Ours	**91.5** (1.8)	**87.4** (0.5)	**91.4** (0.6)	**57.6** (5.5)	**62.2** (6.0)	**66.1** (4.1)	**Ours**	**60.9** (1.9)	**59.2** (3.0)	**32.1** (2.0)	**67.6**
0	LOTClass♣	71.8	81.7	50.1	50.4	36.5	55.9	LOTClass♣	11.5	9.8	2.5	41.1
	FT	49.1	50.0	49.8	34.4	49.5	31.6	FT	10.0	6.2	0.5	31.2
	LM-BFF (man)	83.5	80.3	78.4	49.7	50.5	49.7	KnPr	15.9	10.3	2.3	46.7
	LM-BFF (D-demo)	82.9	80.7	**81.4**	52.2	53.5	44.0	KnPr (D-demo)	—	—	—	47.0*
	KPT †	78.4	81.9	71.4	37.1	55.3	47.5	KPT †	24.6	11.6	0.8	45.7
	Ours	**86.8**	**83.5**	79.7	**53.7**	**56.2**	**56.7**	**Ours**	**41.3**	**12.2**	**2.8**	**52.5**

Summary

"

- **Prompt learning** with pre-trained language model can obtain promising performance and yield more gains with **retrieval augmentation**
- **Data augmentation (e.g., via prompting)** is powerful for few-shot learning
- Task-specific **inductive bias (Schema)** can help data-efficient knowledge graph construction



Scope: Efficient Knowledge Graph Construction

Data Efficiency: few-shot learning with different types of data

- Prompt learning, Retrieval-augmentation

Model Efficiency: parameter-efficient learning, unified architecture

- Parameter-efficient learning

Inference Efficiency: fast Inference

- FastRE, sequence-to-sequence

Why Model-Efficient Knowledge Graph Construction?

Large-scale pre-trained models

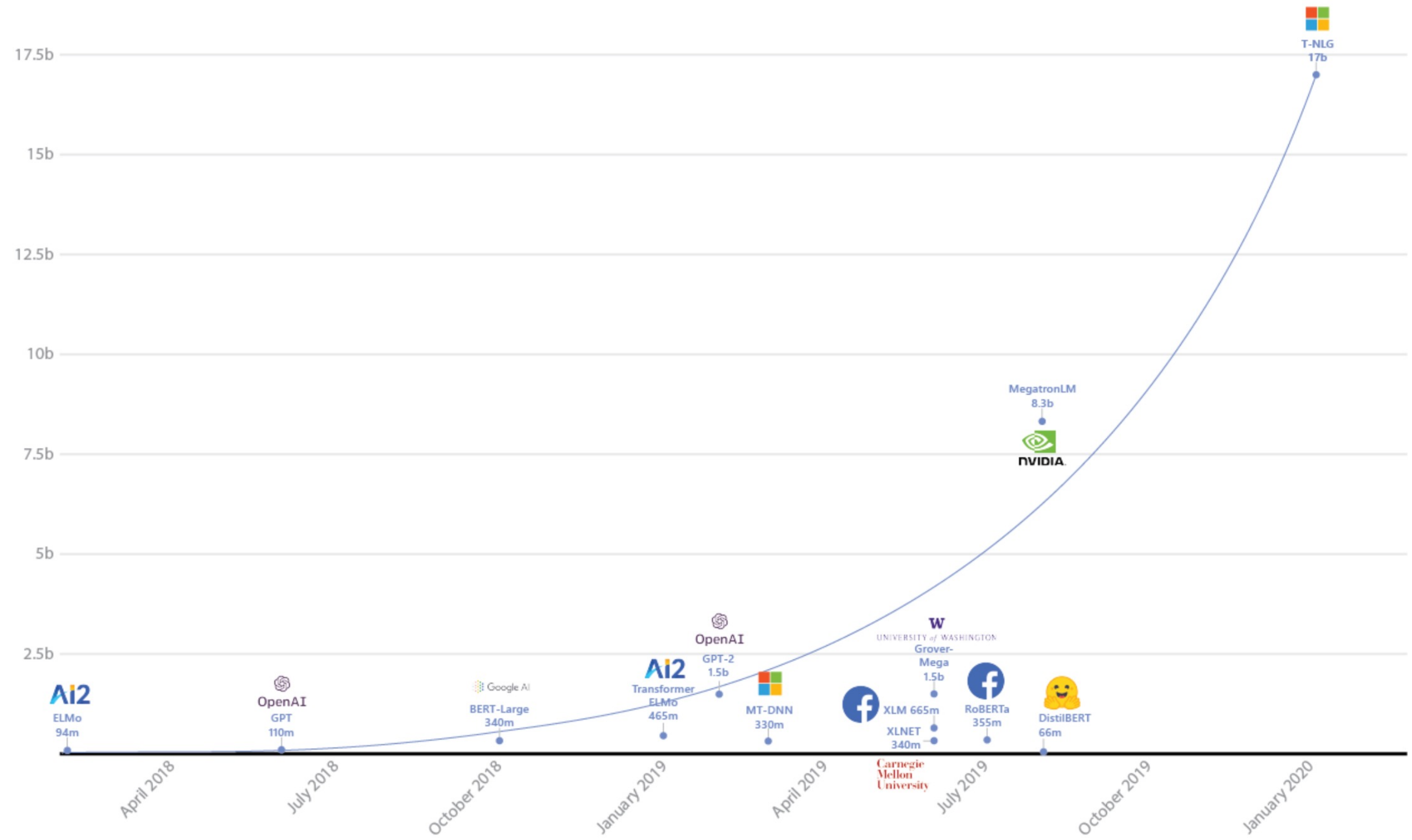


Figure from Turing-NLG

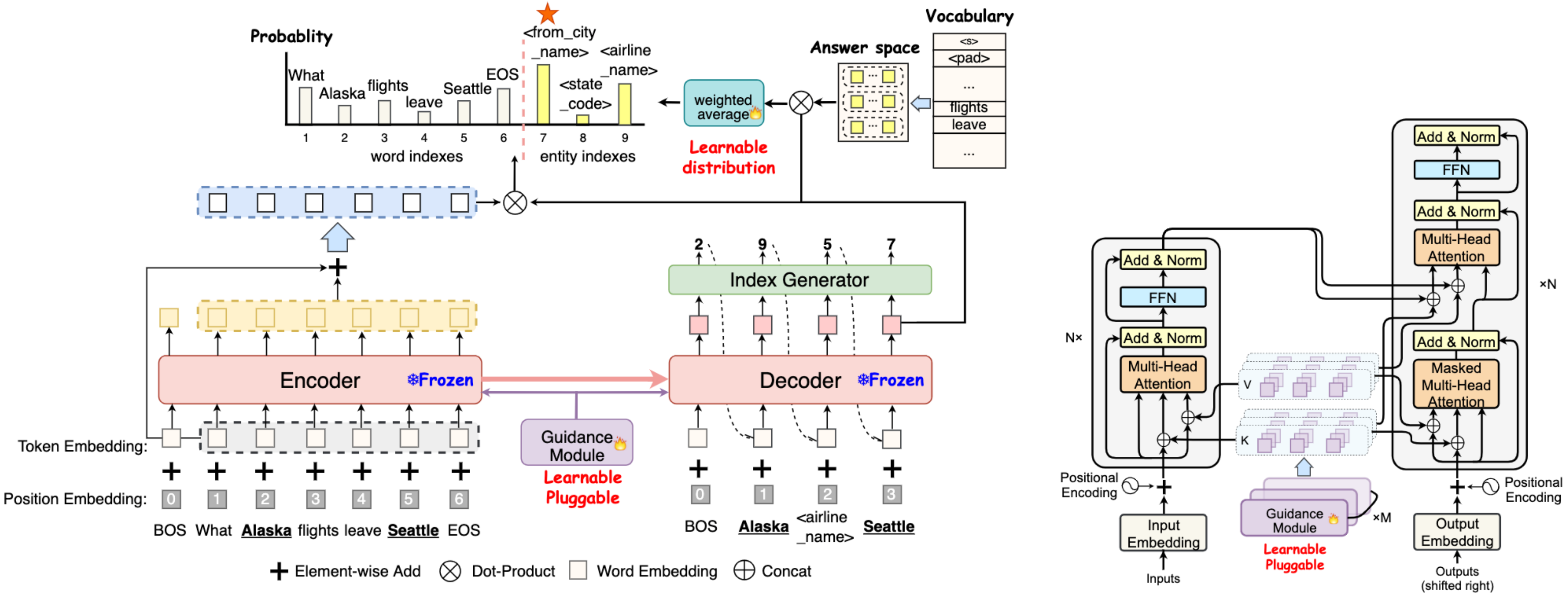
Why Model-Efficient Knowledge Graph Construction?

Efficient transfer learning

LightNER

AACL IJCNLP 2022

Continuous prompts for NER



(a) Overview of LightNER.

(b) Attention with guidance.

Figure 2: Overview of our LightNER framework.

LightNER: A Lightweight Tuning Paradigm for Low-resource NER via Pluggable Prompting (COLING2022)

LightNER

Code is available in https://github.com/zjunlp/DeepKE/tree/main/example/ner/few-shot

Source	Methods	*MIT Movie*						*MIT Restaurant*						*ATIS*		
		10	20	50	100	200	500	10	20	50	100	200	500	10	20	50
None	LC-BERT	25.2	42.2	49.6	50.7	59.3	74.4	21.8	39.4	52.7	53.5	57.4	61.3	44.1	76.7	90.7
	LC-BART	10.2	27.5	44.2	47.5	54.2	64.1	6.3	8.5	51.3	52.2	56.3	60.2	42.0	72.7	87.5
	Template	37.3	48.5	52.2	56.3	62.0	74.9	46.0	57.1	58.7	60.1	62.8	65.0	71.7	79.4	92.6
	BERT-MRC†	18.7	48.3	55.5	62.5	80.2	82.1	12.3	37.1	53.5	63.9	65.5	70.4	35.3	63.2	90.2
	LightNER	**41.7**	**57.8**	**73.1**	**78.0**	**80.6**	**84.8**	**48.5**	**58.0**	**62.0**	**70.8**	**75.5**	**80.2**	**76.3**	**85.3**	**92.8**
CoNLL03	Neigh.Tag.	0.9	1.4	1.7	2.4	3.0	4.8	4.1	3.6	4.0	4.6	5.5	8.1	2.4	3.4	5.1
	Example.	29.2	29.6	30.4	30.2	30.0	29.6	25.2	26.1	26.8	26.2	25.7	25.1	22.9	16.5	22.2
	MP-NSP	36.4	36.8	38.0	38.2	35.4	38.3	46.1	48.2	49.6	49.6	50.0	50.1	71.2	74.8	76.0
	LC-BERT	28.3	45.2	50.0	52.4	60.7	76.8	27.2	40.9	56.3	57.4	58.6	75.3	53.9	78.5	92.2
	LC-BART	13.6	30.4	47,8	49.1	55.8	66.9	8.8	11.1	42.7	45.3	47.8	58.2	51.3	74.4	89.9
	Template	42.4	54.2	59.6	65.3	69.6	80.3	53.1	60.3	64.1	67.3	72.2	75.7	77.3	88.9	93.5
	BERT-MRC†	20.2	50.8	56.3	62.9	81.5	82.3	15.8	39.5	54.8	65.8	68.8	73.5	40.5	66.7	91.8
	LightNER	**62.9**	**75.6**	**78.8**	**82.2**	**84.5**	**85.7**	**58.1**	**67.4**	**69.5**	**73.7**	**78.4**	**81.1**	**86.9**	**89.4**	**93.9**

Table 1: Model performance (F1 score) in the cross-domain low-resource setting. "†" indicates that we rerun their public code in this setting. All of our experiments and baselines adopt large version of LMs.

Low-high Layer vs. High-low Layer

Guidance module applied to **higher layers of LMs can better stimulate knowledge** from PLMs for downstream tasks more efficiently

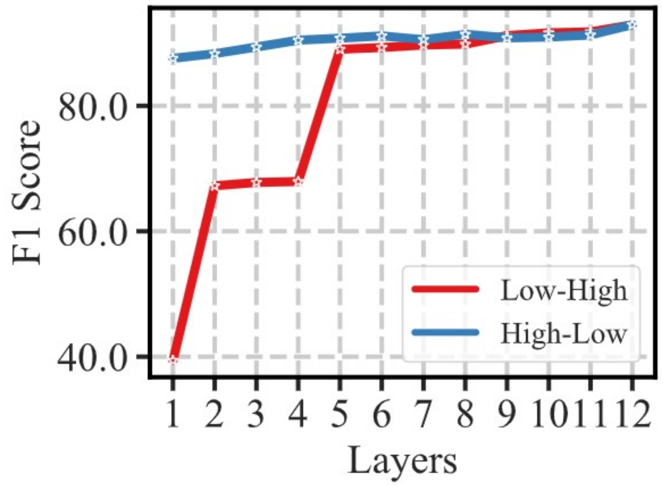
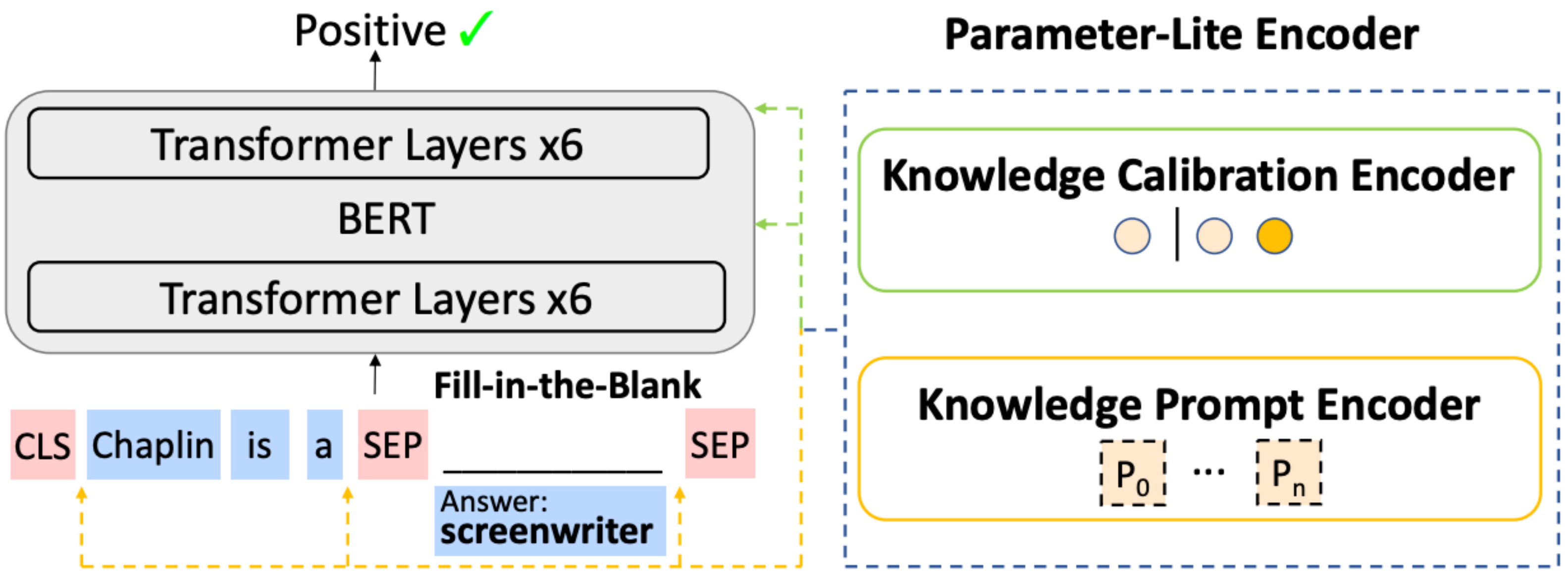


Figure 3: Performances on CoNLL03 as the layers of guidance module varies.

Parameter-efficient Knowledge Graph Completion

Reformulating KG completion as a "fill-in-the-blank" with parameter-lite encoder



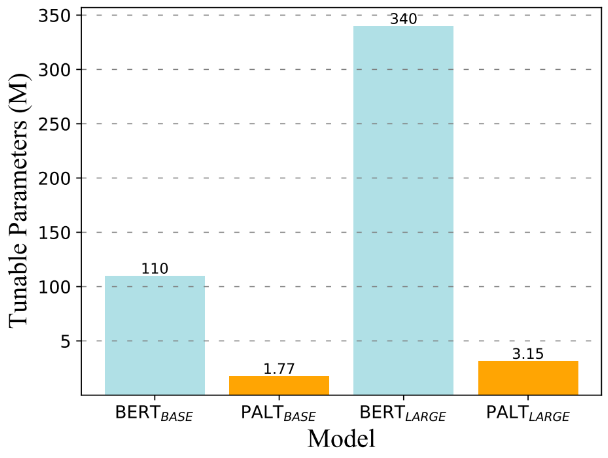


Figure 2: Compare the number of tunable parameters of PALT and BERT.

PALT: Parameter-Lite Transfer of Language Models for Knowledge Graph Completion (EMNLP2022)

Model-efficient Knowledge Graph Construction

Any other solution?

Model-efficient Knowledge Graph Construction

AACL IJCNLP 2022

Unified one model for **ALL**

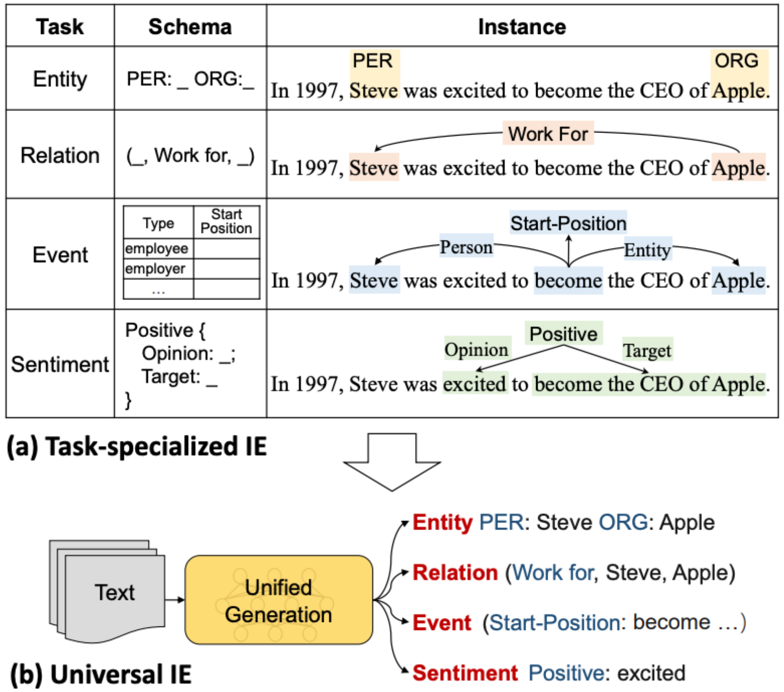


Figure 1: From (a) Task-specialized IE: different tasks, different structures, different schemas to (b) Universal IE: unified modeling via structure generation.

```
(
  (Spot Name: Info Span
    (Asso Name: Info Span)
    (Asso Name: Info Span)
  )
)
```

(a) Structured extraction language (SEL) for Universal IE.

```
(
  (person: Steve
    (work for: Apple)
  )
  (start-position: became
    (employee: Steve)
    (employer: Apple)
    (time: 1997)
  )
  (organization: Apple)
  (time: 1997)
)
```

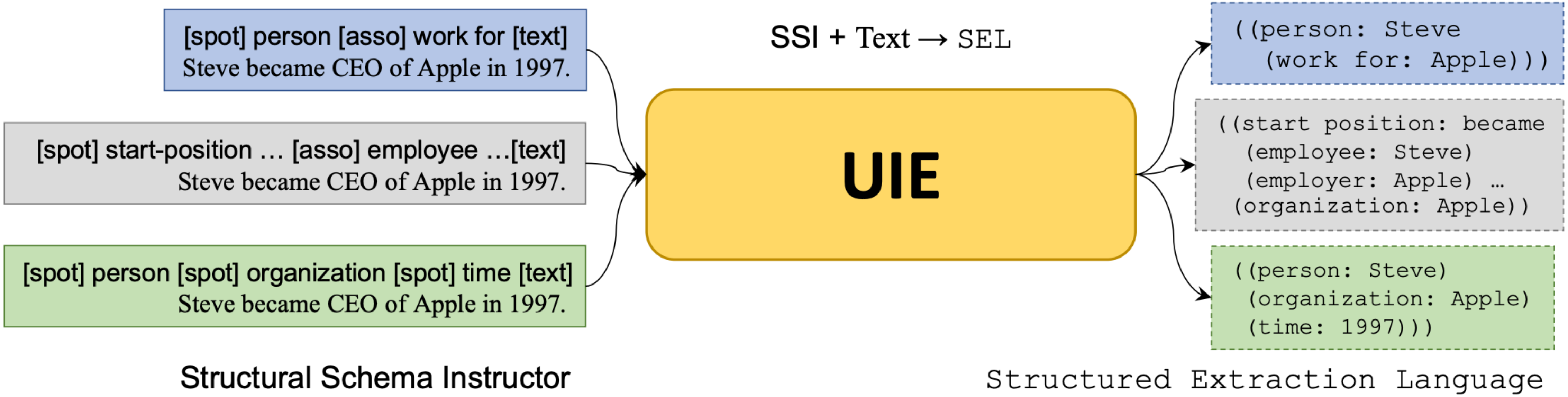
(b) The SEL representation of the extraction structure of "Steve became CEO of Apple in 1997.", where the relation structure is marked blue, the event structure is marked red, and the rest are entities.

Figure 2: Illustrations of structured extraction language.

Model-efficient Knowledge Graph Construction

AACL IJCNLP 2022

Unified one model for **ALL**



Unified Structure Generation for Universal Information Extraction (ACL2022)

Model-efficient Knowledge Graph Construction

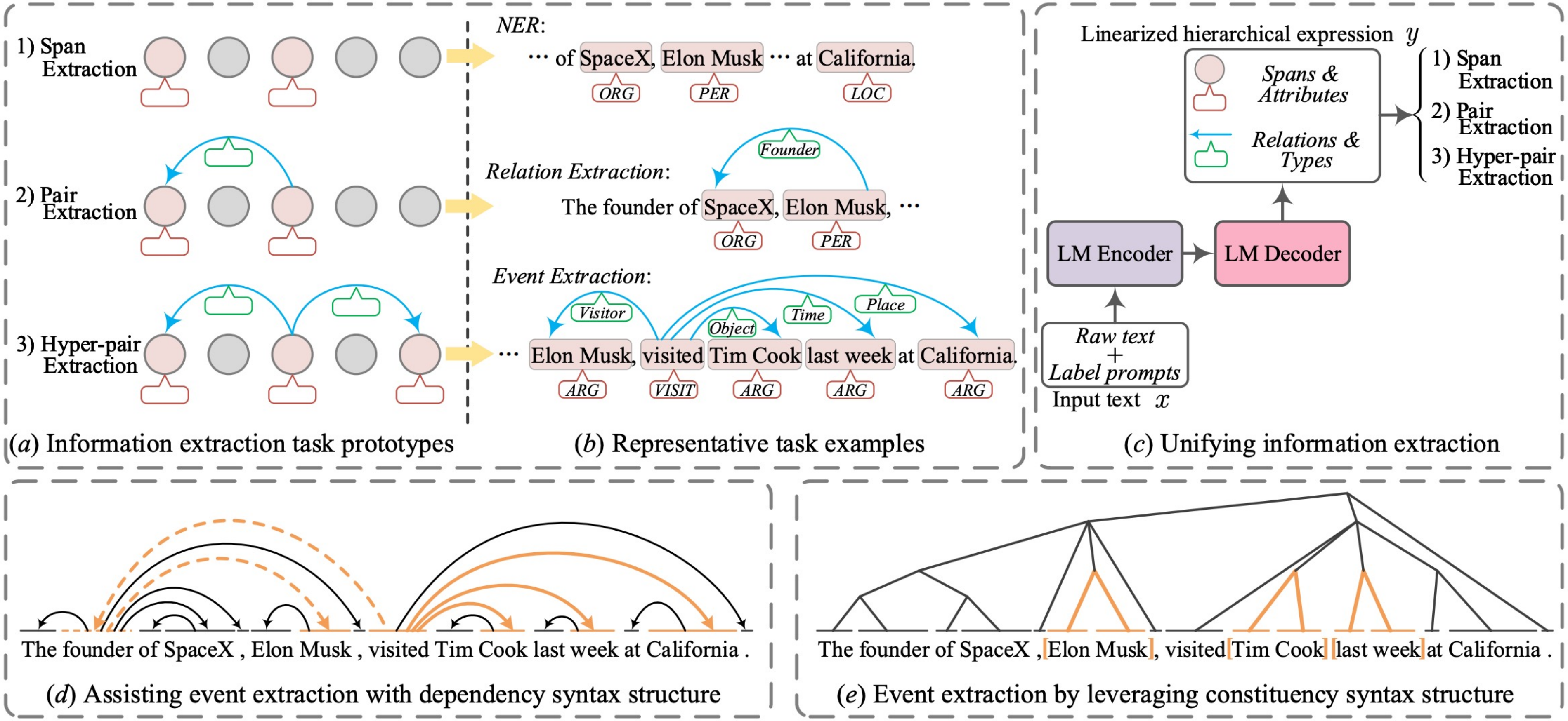
Unified one model for **ALL**

	Model	1-Shot	5-Shot	10-Shot	AVE-S	1%	5%	10%	AVE-R
Entity (CoNLL03) Ent-F1	T5-v1.1-base	12.73	30.17	58.89	33.93	75.74	85.71	87.70	83.05
	Fine-tuned T5-base	24.93	54.85	65.31	48.36	78.51	87.67	88.91	85.03
	UIE-base w/o SSI	43.52	64.76	72.47	60.25	81.91	**88.41**	**89.84**	86.72
	UIE-base	**46.43**	**67.09**	**73.90**	**62.47**	**82.84**	88.34	89.63	**86.94**
Relation (CoNLL04) Rel-S F1	T5-v1.1-base	2.35	7.99	25.98	12.11	6.08	32.38	41.87	26.78
	Fine-tuned T5-base	4.24	28.16	41.44	24.61	12.89	37.75	49.95	33.53
	UIE-base w/o SSI	13.21	40.35	49.47	34.34	24.21	48.70	56.59	43.17
	UIE-base	**22.05**	**45.41**	**52.39**	**39.95**	**30.77**	**51.72**	**59.18**	**47.22**
Event Trigger (ACE05-Evt) Evt Tri F1	T5-v1.1-base	19.40	43.35	50.57	37.77	25.59	49.47	57.18	44.08
	Fine-tuned T5-base	30.18	48.31	51.27	43.25	31.08	51.16	57.76	46.67
	UIE-base w/o SSI	32.07	48.11	51.00	43.73	32.71	53.20	59.26	48.39
	UIE-base	**38.14**	**51.21**	**53.23**	**47.53**	**41.53**	**55.70**	**60.29**	**52.51**
Event Argument (ACE05-Evt) Evt Arg F1	T5-v1.1-base	2.75	20.21	27.53	16.83	3.59	21.53	30.90	18.67
	Fine-tuned T5-base	6.96	25.07	30.96	21.00	7.39	24.97	33.90	22.09
	UIE-base w/o SSI	9.31	23.99	30.31	21.20	9.57	27.25	34.18	23.67
	UIE-base	**11.88**	**27.44**	**33.64**	**24.32**	**12.80**	**30.43**	**36.28**	**26.50**
Sentiment (16res) Rel-S F1	T5-v1.1-base	0.04	2.11	12.66	4.94	3.50	27.08	45.97	25.52
	Fine-tuned T5-base	6.55	21.06	29.92	19.18	18.72	39.63	51.65	36.67
	UIE-base w/o SSI	7.79	17.77	32.07	19.21	19.14	42.76	53.44	38.45
	UIE-base	**10.50**	**26.24**	**39.11**	**25.28**	**24.24**	**49.31**	**57.61**	**43.72**

Unified Structure Generation for Universal Information Extraction (ACL2022)

Model-efficient Knowledge Graph Construction

Unified one model for **ALL**

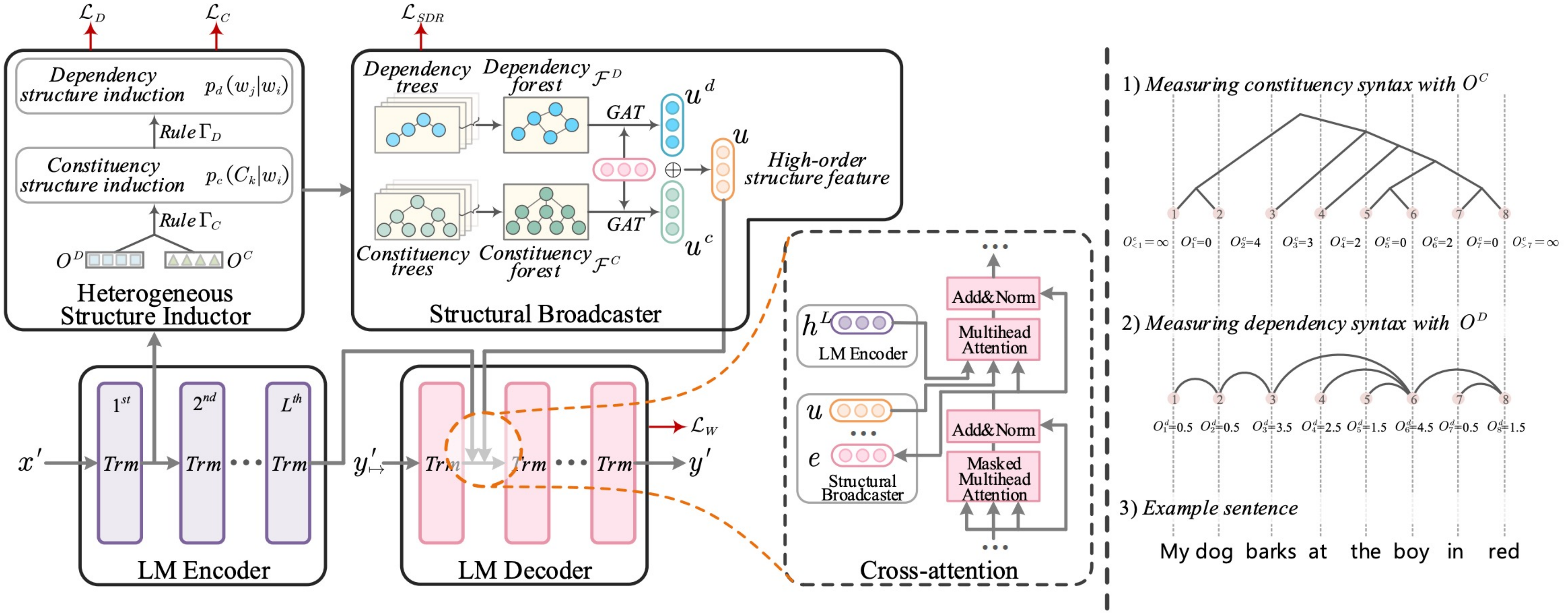


(a) Information extraction task prototypes (b) Representative task examples (c) Unifying information extraction

(d) Assisting event extraction with dependency syntax structure (e) Event extraction by leveraging constituency syntax structure

Model-efficient Knowledge Graph Construction

Unified one model for **ALL**

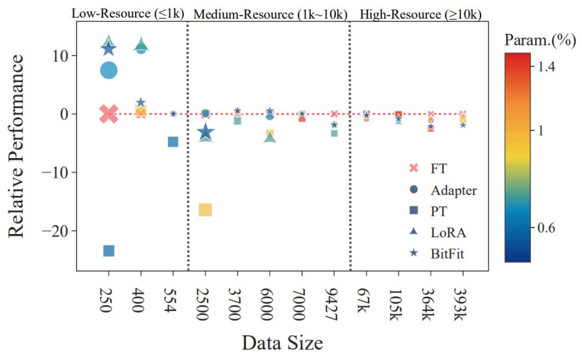


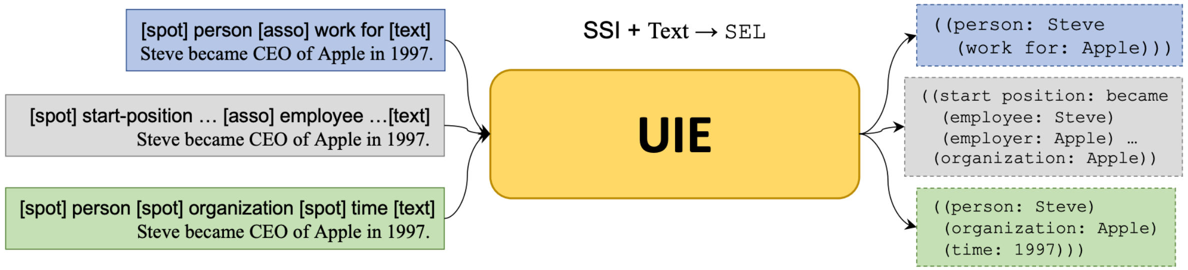
(*a*) Unsupervised structure-aware post-training

(*b*) Heterogeneous syntax measurements

Summary

- By tuning just a fraction amount of parameters comparing to full model finetuning, **parameter-efficient tuning methods** can achieve performance on par with vanilla finetuning
- Instead of task-specialized modeling, it is beneficial to advocate for a **unifying view** of knowledge graph construction





Scope: Efficient Knowledge Graph Construction

Data Efficiency: few-shot learning with different types of data

- Prompt learning, Retrieval-augmentation

Model Efficiency: parameter-efficient learning, unified architecture

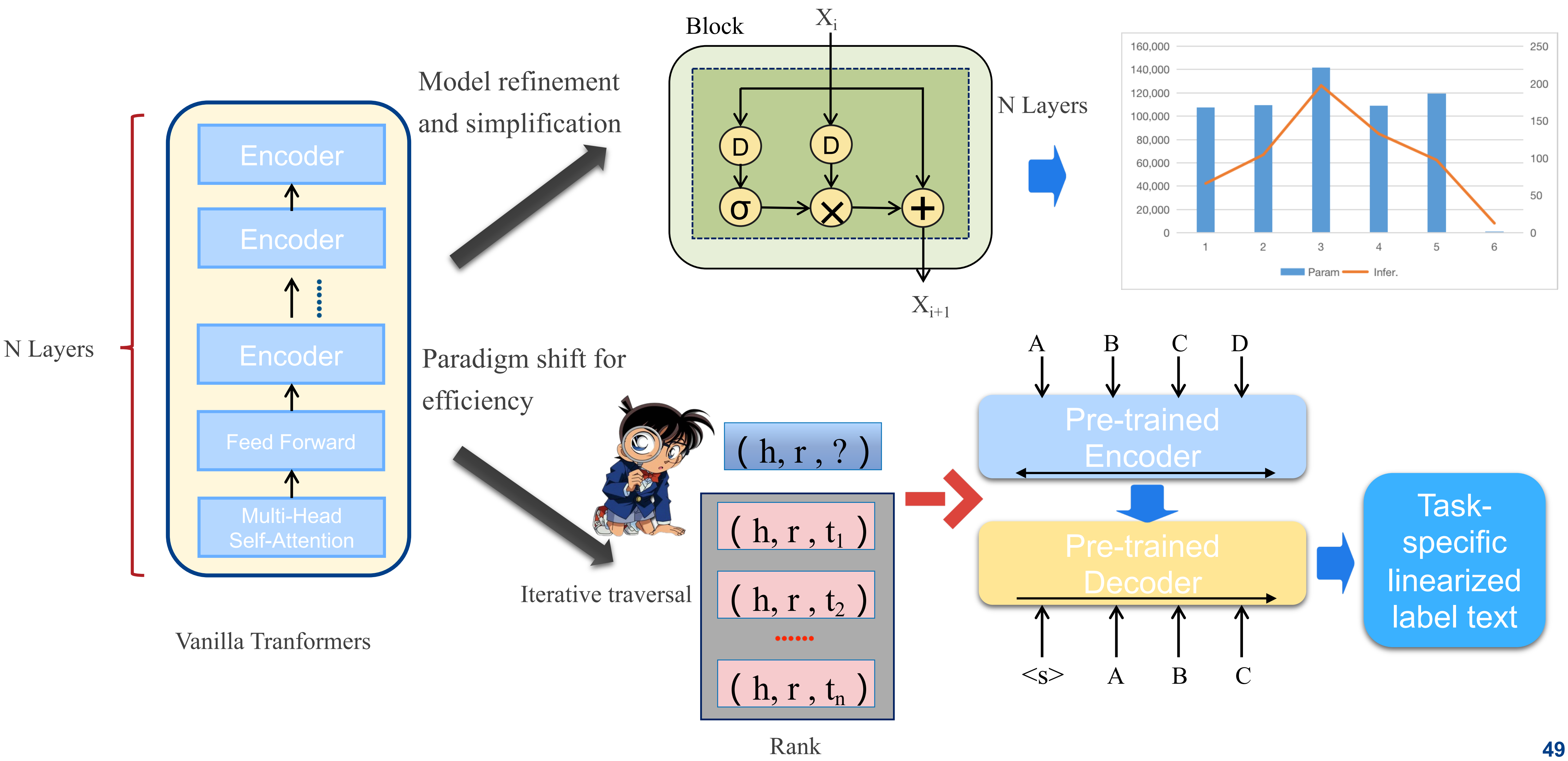
- Parameter-efficient learning

Inference Efficiency: fast Inference

- FastRE, sequence-to-sequence

Very Slow Inference

AACL IJCNLP 2022



Inference-efficient Knowledge Graph Construction

Convolutional encoder, **NOT BERT!**

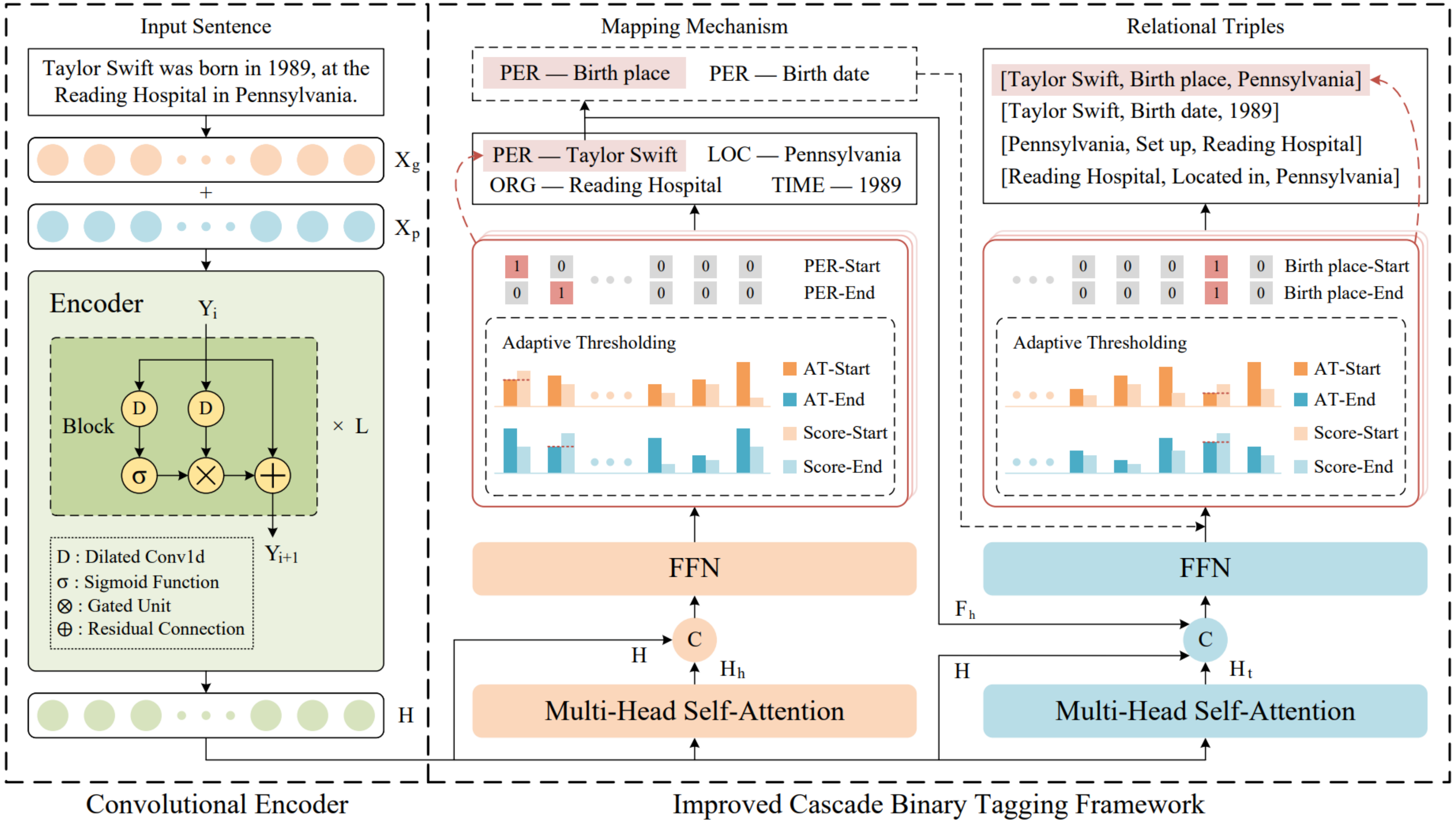


Figure 1: The overall structure of FastRE.

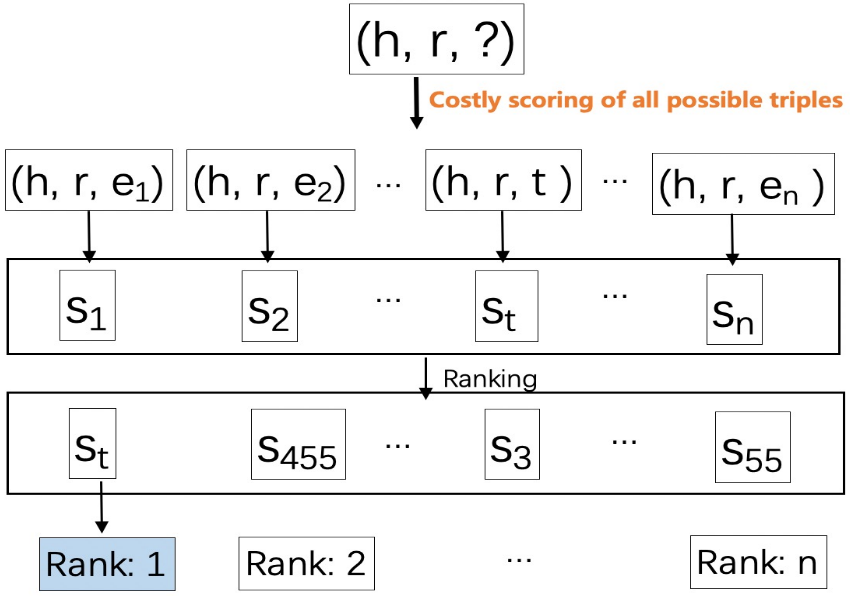
FastRE: Towards Fast Relation Extraction with Convolutional Encoder and Improved Cascade Binary Tagging Framework (IJCAI2022)

Any other solution?

Efficient Knowledge Graph Completion

1. Traditional KGE models embed the entities and relations into a vector space and obtain the predicted triples by leveraging **pre-defined scoring functions** to those vectors.

2. A discrimination strategy has to **costly scoring of all possible triples** in inference and suffer from the instability of negative sampling.



For One Triple	Method	Complexity	Time under RTX 3090
Training	TransE	$O(k+1)$	0.08ms
	KG-BERT	$O(\|d\|^2 \times (k+1))$	72ms
	GenKGC	$O(\|d\|^2)$	2.35ms
Inference	TransE	$O(\|\mathcal{E}\|)$	0.02s
	KG-BERT	$O(\|d\|^2 \times \|\mathcal{E}\|)$	10100s
	GenKGC	$O(\|d\|^2 \times \|d\|^k)$	0.96s

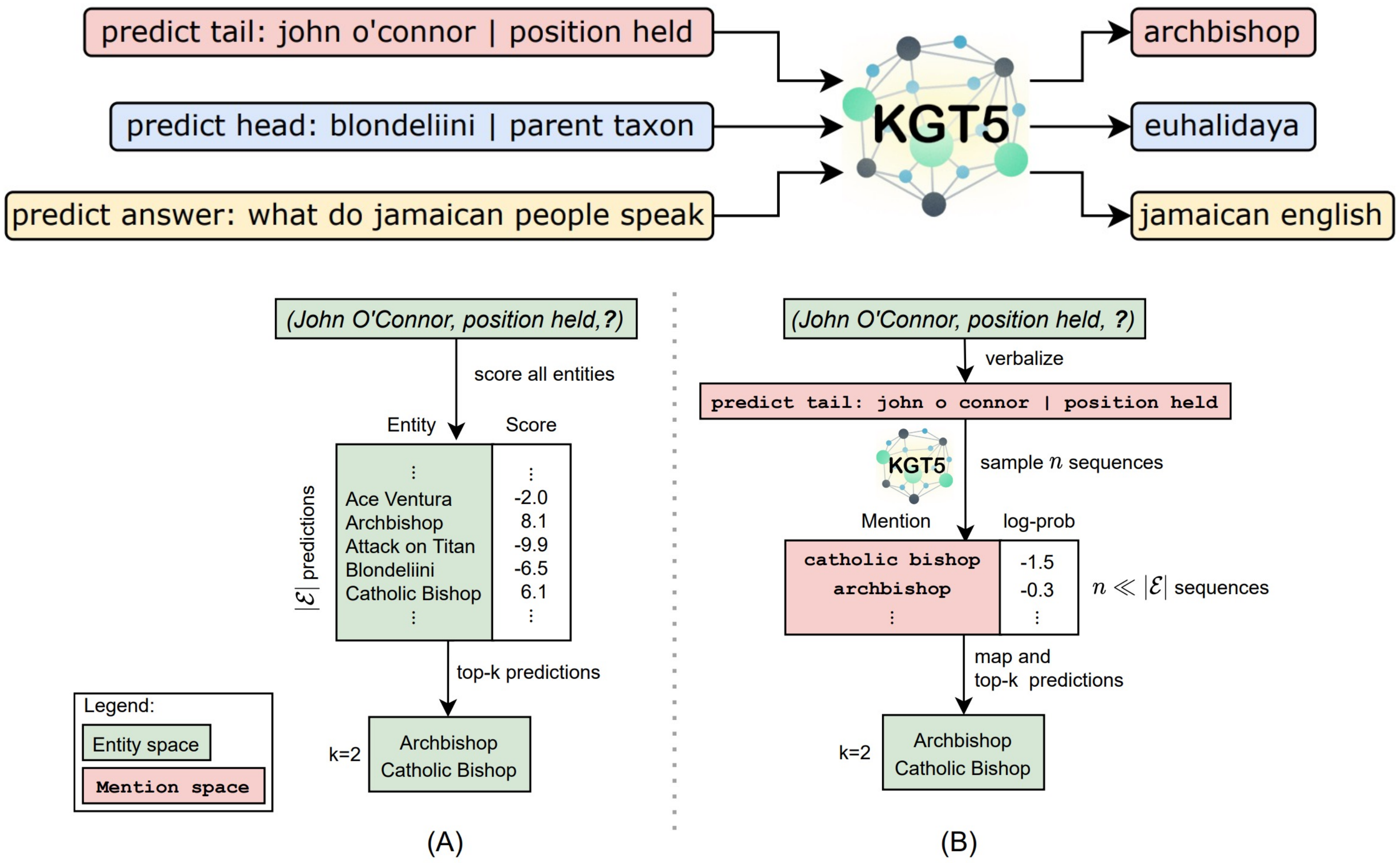
Our Motivation

How to complete Knowledge Graph with PLMs **efficiently**?

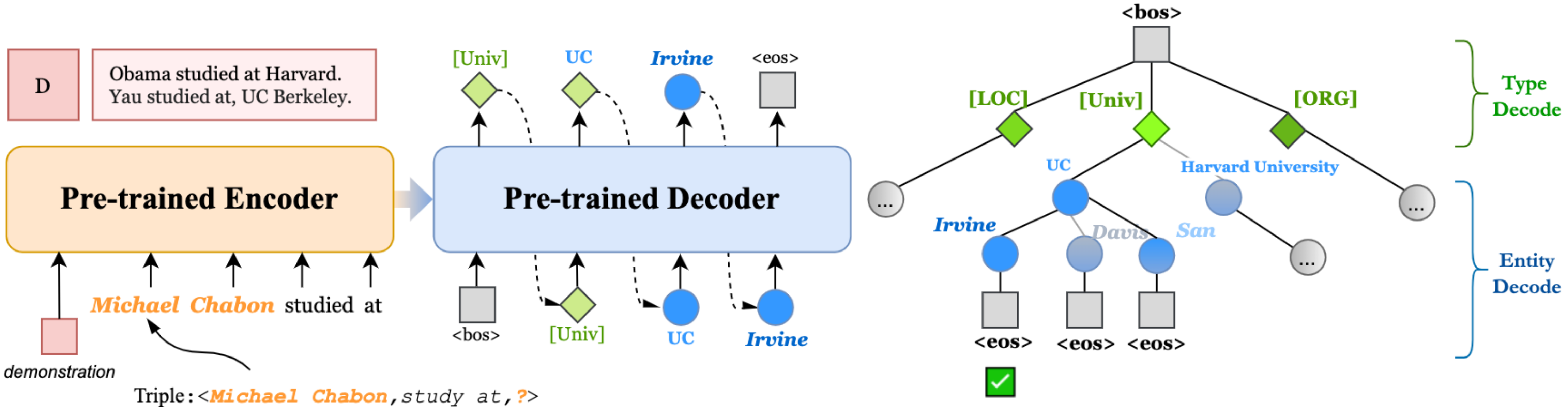
One solution: *Fast Inference with Generation*

KGT5

Encoder-decoder Transformer model can serve as a scalable and versatile model



GenKGC



$$p_\theta(y \mid x) = \prod_{i=1}^{|c|} p_\theta\left(z_i \mid z_{<i}, x\right) \prod_{i=|c|+1}^{N} p_\theta\left(y_i \mid y_{<i}, x\right), \qquad (1)$$

$$\mathcal{L} = -\log p_\theta(y \mid x)$$

GenKGC

AACL IJCNLP 2022

Benchmark and real-world datasets

Datasets:

- WN18RR
- FB15k-237
- OpenBG500 (A New KG)

Code for reproducibility:
https://github.com/zjunlp/PromptKG/tree/main/research/GenKGC

Table 3: Summary statistics of benchmark datasets.

Dataset	# Ent	# Rel	# Train	# Dev	# Test
WN18RR	40,943	11	86,835	3,034	3,134
FB15k-237	14,541	237	272,115	17,535	20,466
OpenBG500	269,658	500	1,242,550	5,000	5,000

OpenBG: Large Scale Open Business Knowledge Graph Evaluation Benchmark Leaderboard

AliOpenKG 2022-03-08 417 98

Apply for dataset New a notebook

Content Notebook Comment Leaderboard Submission

Description

OpenBG Benchmark is a large-scale open business knowledge graph evaluation benchmark, which includes multiple sub-datasets and sub-tasks. The dataset is built on the basis of OpenBG. OpenBG is an open business knowledge graph that utilizes a unified Schema covering multi-modal datasets in large scale, which contains millions of products and consumer demand provided by Alibaba Knowledge Engine Group and Zhejiang University. The goal of OpenBG is to use open business knowledge to discover the value of the social economy, promote cross-disciplinary research in the fields of digital commerce and digital economy, and serve the national strategic needs of the healthy development of digital economy.

Website: https://tianchi.aliyun.com/OpenBG
Github: https://github.com/OpenBGBenchmark

https://github.com/OpenBGBenchmark/OpenBG

GenKGC

AACL IJCNLP 2022

Method	WN18RR			FB15k-237			OpenBG500		
	Hits@1	Hits@3	Hits@10	Hits@1	Hits@3	Hits@10	Hits@1	Hits@3	Hits@10
Graph embedding approach									
TransE [2] ◇	[illegible]	[illegible]	0.532	0.198	0.376	0.441	0.207	0.340	0.513
DistMult [13] ◇	[illegible]	0.470	0.504	0.199	0.301	0.446	0.049	0.088	0.216
ComplEx [11] ◇	0.409	0.469	0.530	0.194	0.297	0.450	0.053	0.120	0.266
RotatE [9]	0.428	0.492	0.571	0.241	0.375	0.533	-	-	-
TuckER [1]	0.443	0.482	0.526	0.226	0.394	0.544	-	-	-
ATTH [4]	0.443	0.499	0.486	0.252	0.384	0.549	-	-	-
Textual enco[illegible]									
KG-BERT [14]	0.041	0.302	0.524	[illegible]	[illegible]	[illegible]	0.023	0.049	0.241
StAR [12]	[illegible]	[illegible]	0.709	0.205	[illegible]	0.482	-	-	-
GenKGC	[illegible]	[illegible]	0.535	[illegible]	[illegible]	[illegible]	0.203	0.280	0.351

260M Para.

100,100s

110M Para.

0.96s

From Discrimination to Generation: Knowledge Graph Completion with Generative Transformer(WWW2022 poster)

Generative Knowledge Graph Construction

Public Repository : https://github.com/zjunlp/Generative_KG_Construction_Papers

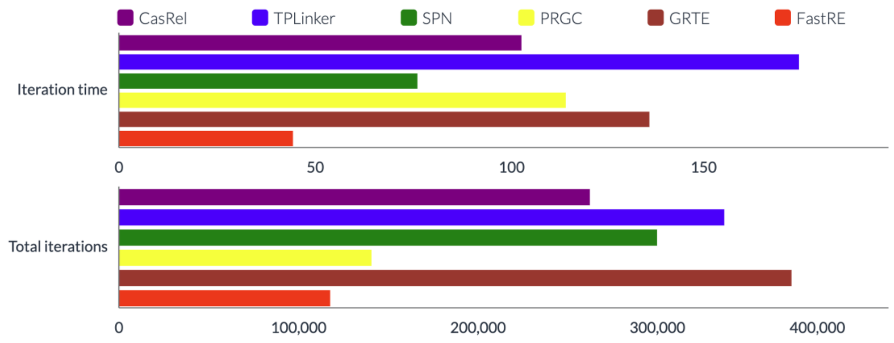
- Generative KGC Taxonomy
  - Generation Target
    - Copy-based Sequence: CopyRE (Zeng et al., 2018), CopyRRL (Zeng et al., 2019), CopyMTL (Zeng et al., 2020), TEMPGEN (Huang et al., 2021), Seq2rel (Giorgi et al., 2022)
    - Structure-based Sequence: Seq2Seq4ATE (Ma et al., 2019), Nested-seq (Straková et al., 2019), CGT (Zhang et al., 2021b; Ye et al., 2021), PolicyIE (Ahmad et al., 2021), Text2Event (Lu et al., 2021), HySPA (Ren et al., 2021), REBEL (Cabot and Navigli, 2021), SQUIRE (Bai et al., 2022), GenKGC (Xie et al., 2022), EPGEL (Lai et al., 2022), HuSe-Gen (Saha et al., 2022), UIE (Lu et al., 2022), DEEPSTRUCT (Wang et al., 2022), De-Bias (Zhang et al., 2022b), KGT5 (Saxena et al., 2022), KG-S2S (Chen et al., 2022a)
    - Label-based Sequence: ANL (Athiwaratkun et al., 2020a), GENRE (Cao et al., 2021), TANL (Paolini et al., 2021)
    - Indice-based Sequence: PNDec (Nayak and Ng, 2020), SEQ2SEQ-PTR (Rongali et al., 2020), GRIT (Du et al., 2021a), UGF for NER (Yan et al., 2021b), UGF for ABSA (Yan et al., 2021a)
    - Blank-based Sequence: COMET (Bosselut et al., 2019), BART-Gen (Li et al., 2021b), GTT (Du et al., 2021b), DEGREE (Hsu et al., 2022), ClarET (Zhou et al., 2022), GTEE (Liu et al., 2022), X-GEAR (Huang et al., 2022), PAIE (Ma et al., 2022)
  - Generation Tasks
    - Named Entity Recognition: Nested-seq (Straková et al., 2019), ANL (Athiwaratkun et al., 2020a), TANL (Paolini et al., 2021), HySPA (Ren et al., 2021), UGF for NER (Yan et al., 2021b), DEEPSTRUCT (Wang et al., 2022), De-Bias (Zhang et al., 2022b), UIE (Lu et al., 2022)
    - Relation Extraction: CopyRE (Zeng et al., 2018), CopyRRL (Zeng et al., 2019), PNDec (Nayak and Ng, 2020), CopyMTL (Zeng et al., 2020), CGT (Zhang et al., 2021b; Ye et al., 2021), TANL (Paolini et al., 2021), HySPA (Ren et al., 2021), TEMPGEN (Huang et al., 2021), REBEL (Cabot and Navigli, 2021), DEEPSTRUCT (Wang et al., 2022), UIE (Lu et al., 2022), Seq2rel (Giorgi et al., 2022)
    - Event Extraction: CGT (Zhang et al., 2021b; Ye et al., 2021), TANL (Paolini et al., 2021), BART-Gen (Li et al., 2021b), GTT (Du et al., 2021b), GRIT (Du et al., 2021a), Text2Event (Lu et al., 2021), DEGREE (Hsu et al., 2022), ClarET (Zhou et al., 2022), GTEE (Liu et al., 2022), X-GEAR (Huang et al., 2022), DEEPSTRUCT (Wang et al., 2022), PAIE (Ma et al., 2022), UIE (Lu et al., 2022)
    - Entity Linking: GENRE (Cao et al., 2021), EPGEL (Lai et al., 2022)
    - Knowledge Graph Completion: COMET (Bosselut et al., 2019), SQUIRE (Bai et al., 2022), GenKGC (Xie et al., 2022),, HuSe-Gen (Saha et al., 2022), ClarET (Zhou et al., 2022), KGT5 (Saxena et al., 2022), KG-S2S (Chen et al., 2022a)

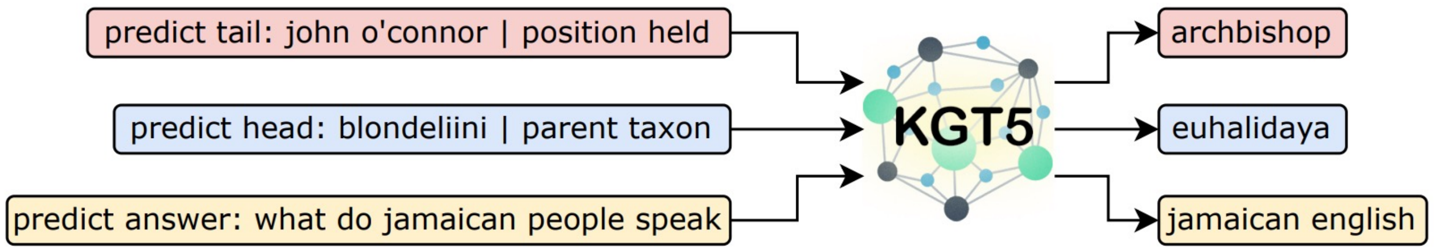
Generative Knowledge Graph Construction: A Review (EMNLP2022)

Summary

❑ **Vanilla encoders (such as CNN)** with several innovations can improve efficiency while also keeping promising performance

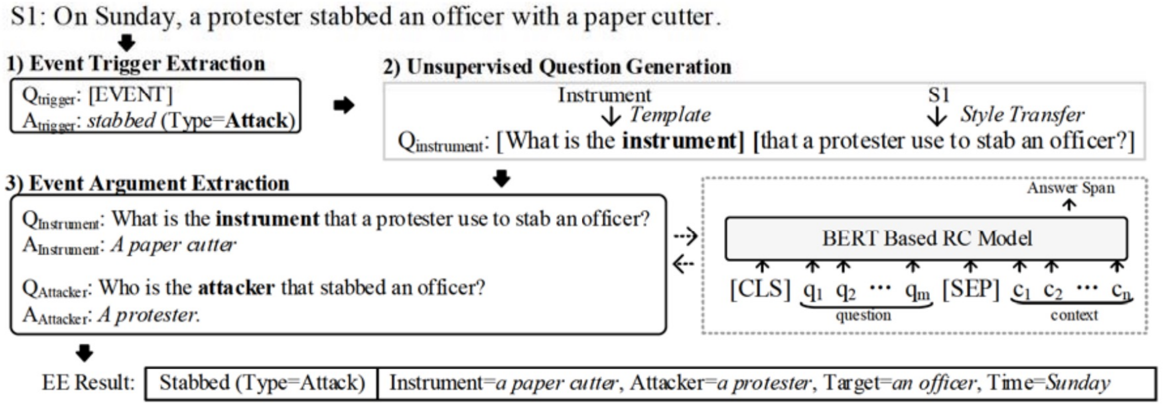
❑ **Paradigm shifting (such as from discrimination to generation)** can help reduce the inference time cost



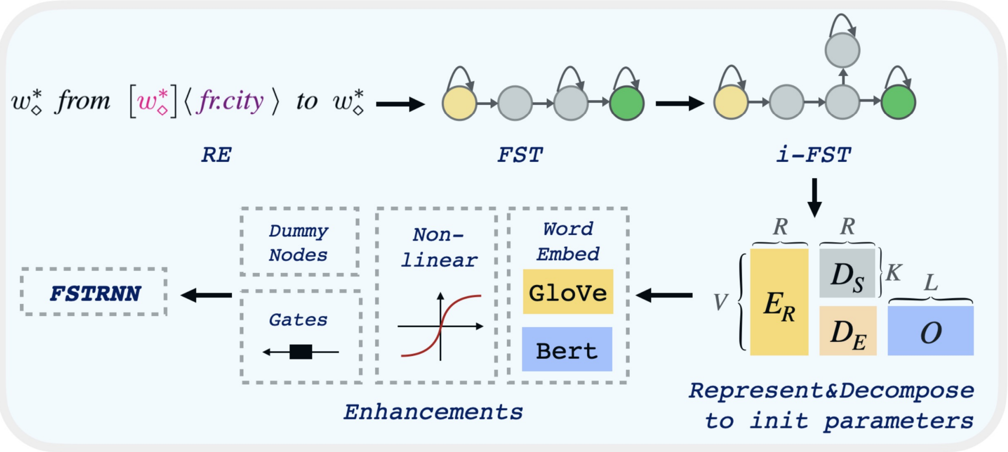


Other Approaches

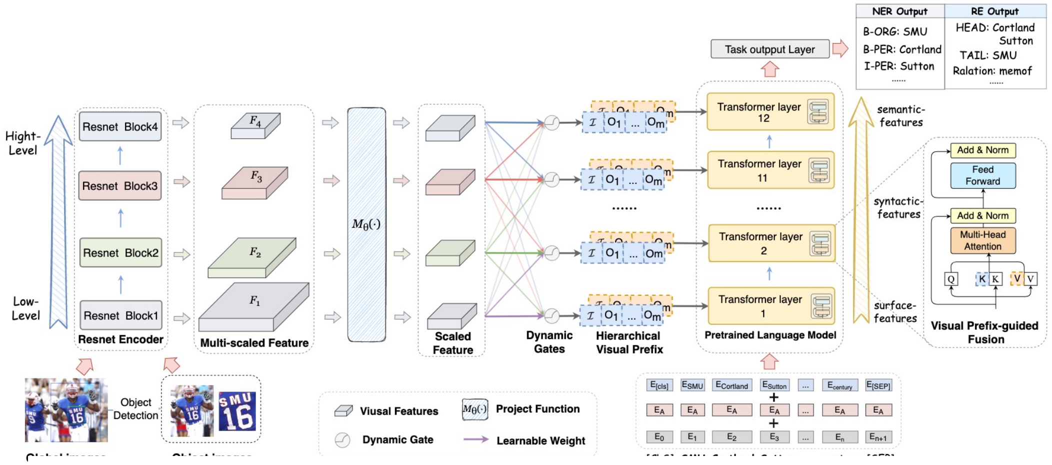
Formulating KE as QA/MRC

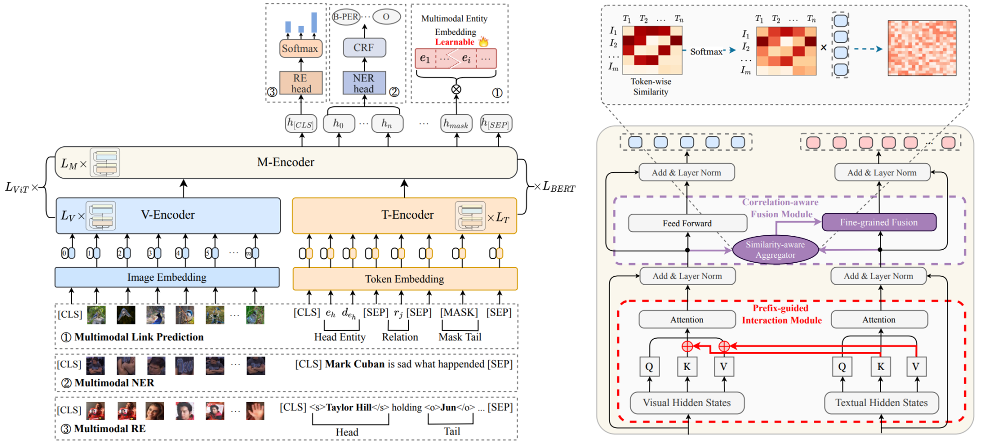


Converting regular expressions into NN



Multimodal enhancement





(a) Unified Multimodal KGC Framework.

(b) Detailed M-Encoder.

Language Model Priming for Cross-Lingual Event Extraction (AAAI2022)
Event Extraction as Machine Reading Comprehension (EMNLP2020)
Neuralizing Regular Expressions for Slot Filling (EMNLP2021)
Good Visual Guidance Make A Better Extractor: Hierarchical Visual Prefix for Multimodal Entity and Relation Extraction (NAACL 2022 Findings)

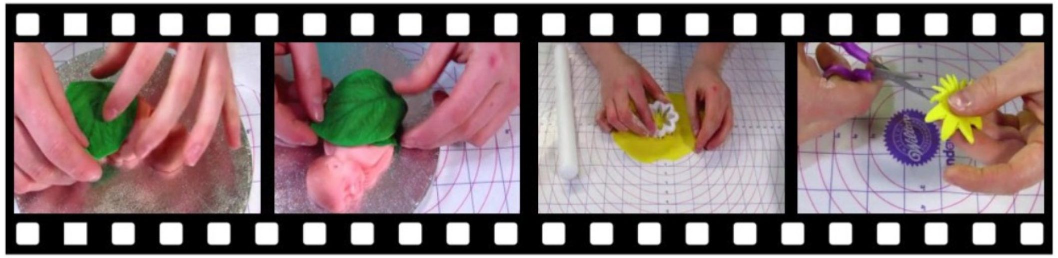
What can we do?

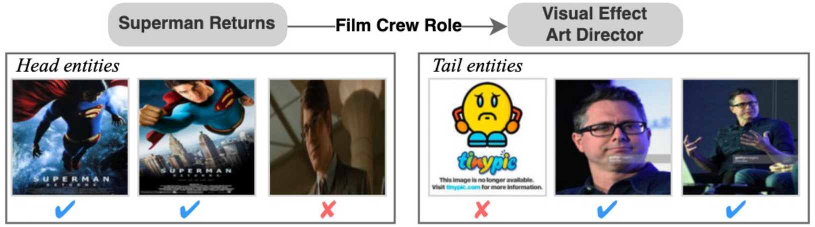
Remaining Issues and Future Works

Remaining Issue1

Efficient **multimodal** Knowledge Graph Construction

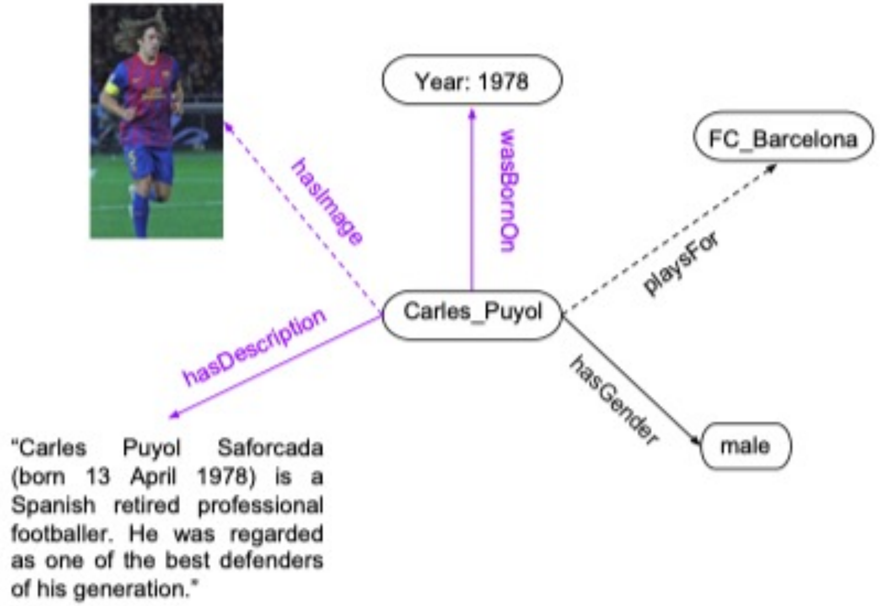
- **When** and **why** visual information is helpful?
- Rich semantic in **dynamic videos**





(a) Examples of Multimodal Link Prediction

(b) Examples of Multimodal Relation Extraction

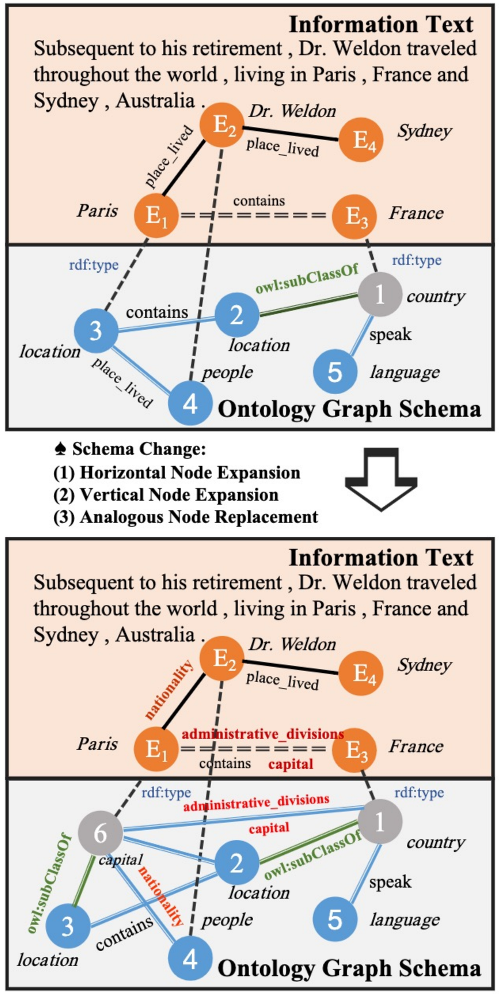


Is Visual Context Really Helpful for Knowledge Graph? A Representation Learning (MM21)
Multi-Modal Knowledge Graph Construction and Application: A Survey 2022

Remaining Issue2

Efficient **lifelong** Knowledge Graph construction

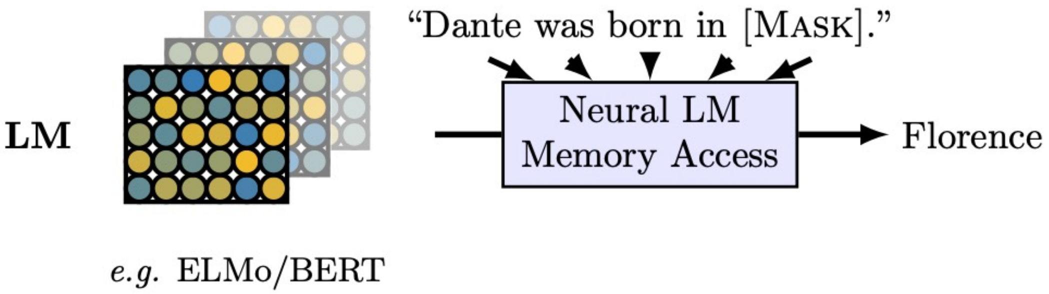
- Rich **connections** between schema
- Schema **self-adapting**, **self-boosting** and **self-correction**



Curriculum-meta Learning for Order-robust Continual Relation Extraction (AAAI2021)
Refining Sample Embeddings with Relation Prototypes to Enhance Continual Relation Extraction (ACL2021)

Remaining Issue3

Manipulating **knowledge in PLM** for efficient KGC



Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let's think step by step.***

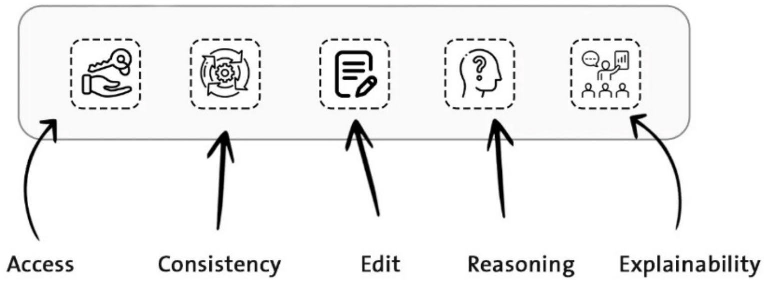
(Output) *There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls.* ✓

- ❑ Use PLM to obtain fact knowledge
- ❑ Can the PLM have the reasoning abilities?

Symbolic | Convert | Hierarchy | Hybrid | Neural

Many remaining issues !

LMs-as-KBs



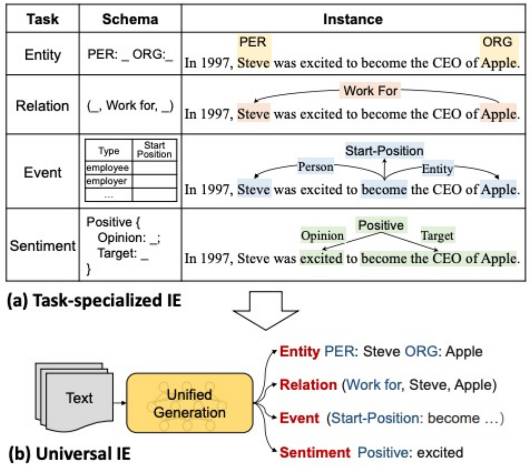
Symbolic VS Neural -> Symbolic + Neural

Knowledgeable or Educated Guess? Revisiting Language Models as Knowledge Bases (ACL2021)
A Review on Language Models as Knowledge Bases 2022

Efficient KGC for Social Good

Efficient NER for information extraction and help **information visualization**

Outlook： Changes of Interactive Objects *Opportunities*



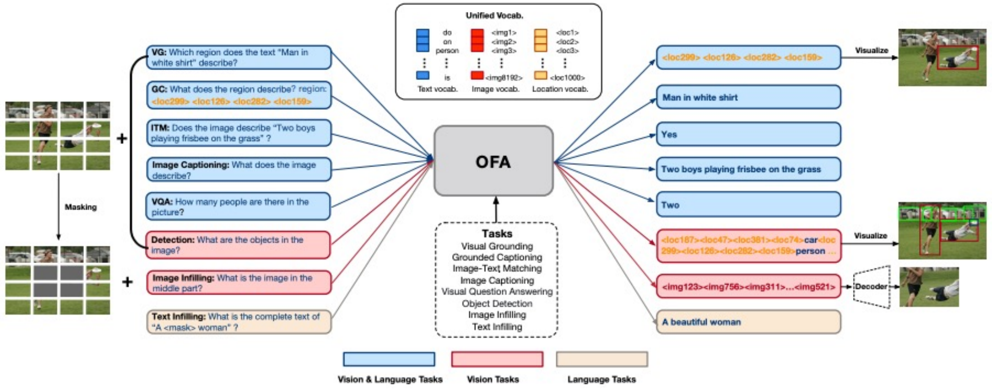
UIE’ ACL22

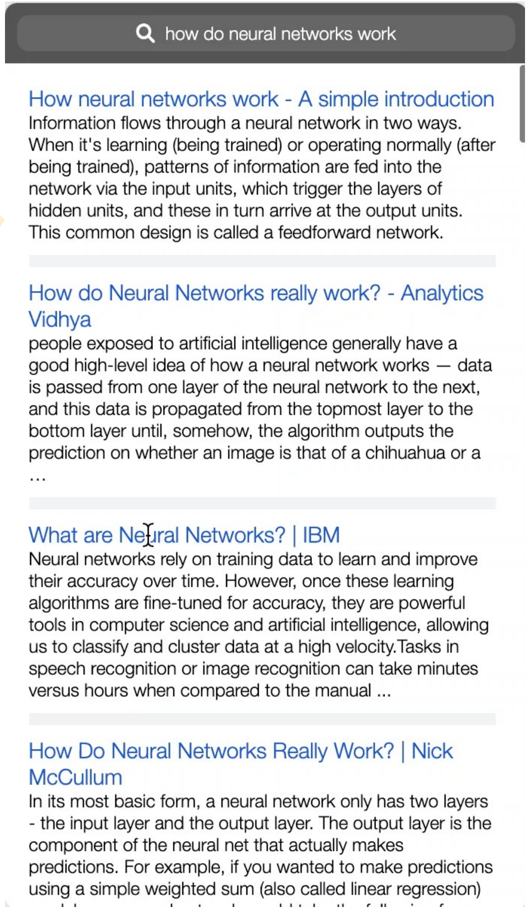
GPT3

With massive text data, **language pre-training**

With multi-modal, multi-type data, **multi-task/modal pre-training**

With the world, environment, **embodied pre-training**





WebGPT

How can we use?

Open-sourced Toolkits

General Tools

OpenPrompt

An Open-Source Framework for Prompt-learning.

OpenDelta

An Open-Source Framework for Paramter-Efficient Tuning (Delta Tuning).

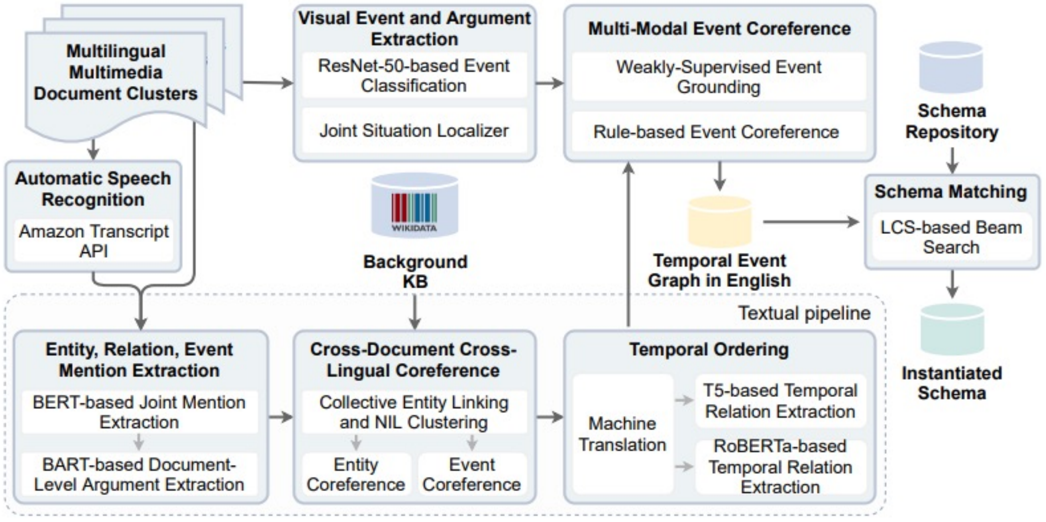
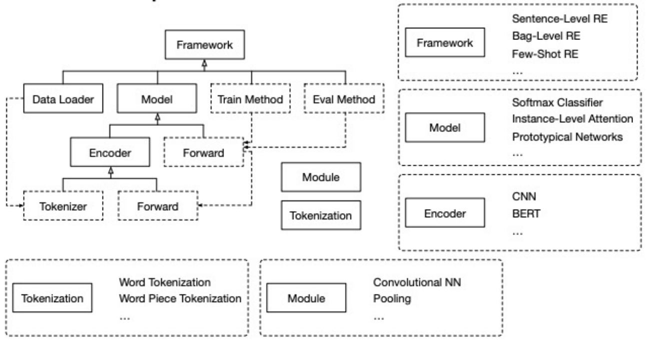


Figure 1: The architecture of RESIN schema-guided information extraction and temporal event tracking system.

RESIN

Architecture of OpenNRE



Example Code

```
rel2id = 'relation->id dictionary'
# Define encoder
sentence_encoder = nrekit.encoder.BERTEncoder(
    max_length=80, pretrain_path='bert pretrain model path')
# Define model
model = nrekit.model.SoftmaxNN(sentence_encoder, len(rel2id), rel2id)
# Define framework
framework = nrekit.framework.SentenceRE(
    train_path='path of training data',
    val_path='path of validation data',
    test_path='path of test data',
    model=model,
    ckpt='path of checkpoint',
    batch_size=64,
    max_epoch=10,
    lr=3e-5,
    opt='bert_adam')
# Train
framework.train_model()
# Test
framework.load_state_dict(torch.load(ckpt)['state_dict'])
result = framework.eval_model(framework.test_loader)
```

OpenNRE

RESIN: A Dockerized Schema-Guided Cross-document Cross-lingual Cross-media Information Extraction and Event Tracking System (NAACL2021 Demo)
OpenNRE: An Open and Extensible Toolkit for Neural Relation Extraction (EMNLP2019)
OpenPrompt: An Open-source Framework for Prompt-learning (ACL2022)

ZS4IE

Public Repository：https://github.com/BBN-E/ZS4IE

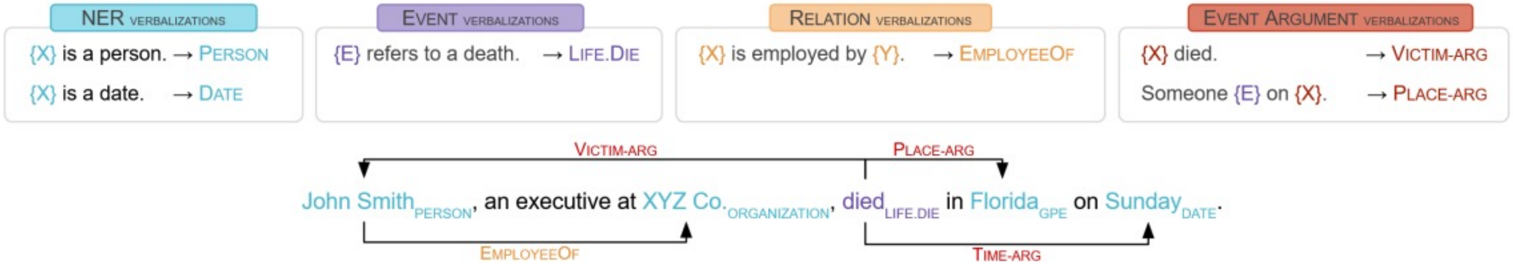


Figure 1: Verbalization templates for a sample schema involving four tasks (from left to right, NER, EE, RE, EAE), with example output (bottom). The schema contains a EmployeeOf relation between Person and Organization entities and a Life.Die event with three argument types (Victim, Place and Time) and Person, Date and GPE entities as fillers. Due to space constraints, at most two verbalizations per task shown.

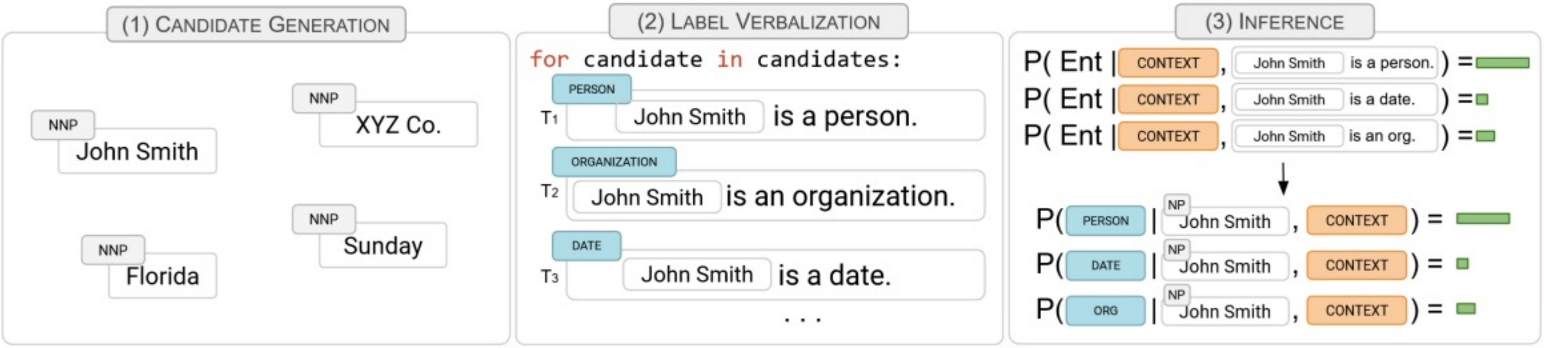


Figure 2: Three steps for entailment-based NER. The steps for the other IE tasks is analogous.

ZS4IE: A toolkit for Zero-Shot Information Extraction with simple Verbalizations (NAACL2022 Demo)

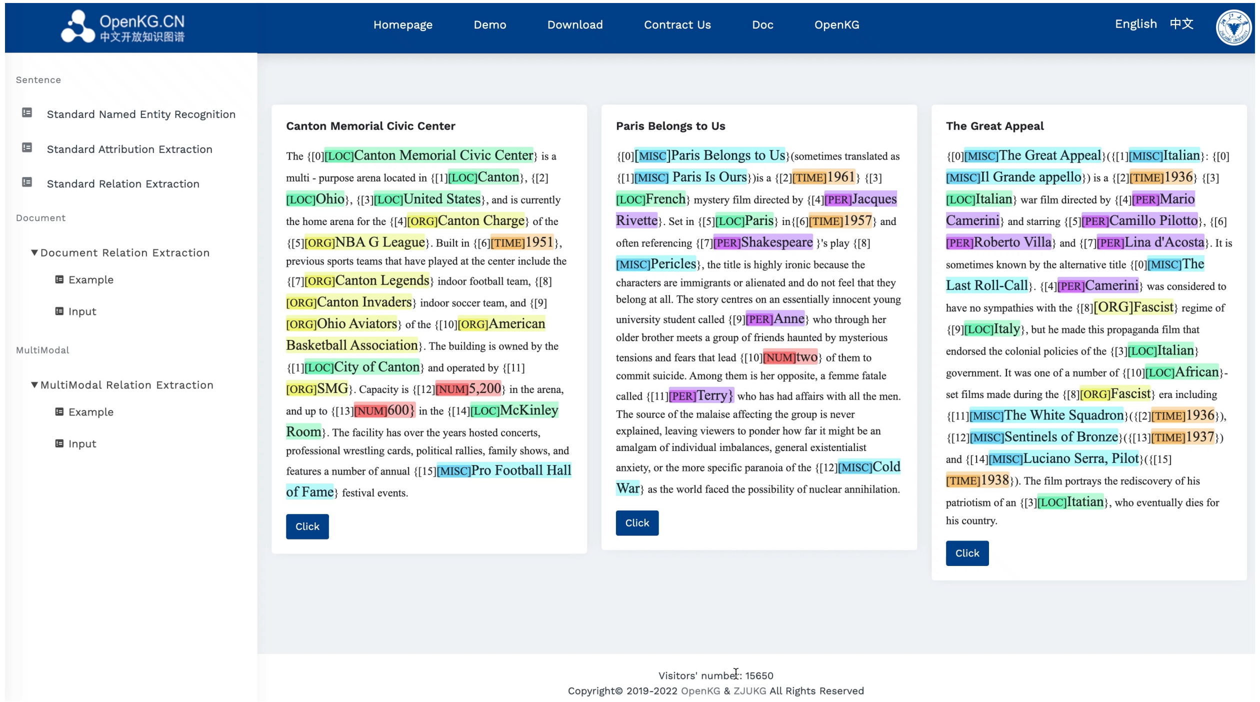
Open-sourced Frameworks

DeepKE is a knowledge extraction toolkit supporting low-resource, document-level and multimodal scenarios.
PromptKG is a prompt learning framework for knowledge graph representation learning and applications.

DeepKE

Knowledge Extraction Tool

PromptKG



https://github.com/zjunlp

DeepKE: A Deep Learning Based Knowledge Extraction Toolkit for Knowledge Base Population (EMNLP 2022 Demo)
PromptKG: A Prompt Learning Framework for Knowledge Graph Representation Learning and Application 2022

DeepKE: Diverse Scenarios

- **Single Sentence**

[LOC]

It was one o'clock when we left **Lauriston Gardens**,
Sherlock Holmes led me meet **Gregson** from **Scotland Yard**.

[PER] [PER] [ORG]

- **Document**

... Elias Brown (May 9, 1793 - July 7, 1857) was a **U.S.** Representative from **Maryland**. Born near **Baltimore**, **Maryland**, Brown attended the common schools ... He died near **Baltimore**, **Maryland**, and is interred in a private cemetery near **Eldersburg**, **Maryland**. ...

Intra-sentence	(**Maryland**, country, **U.S.**)
	(**Baltimore**, located in, **Maryland**)
	(**Eldersburg**, located in, **Maryland**)
Inter-sentence	(**Baltimore**, country, **U.S.**)
	(**Eldersburg**, country, **U.S.**)

- **MultiModal**

Piolo Pascual gives Arci Munoz a kiss as they meet at SIFY

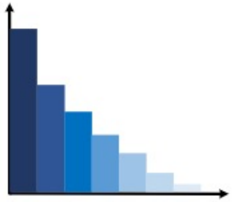
Rel:per/per/couple

DeepKE

Knowledge Extraction Tool

Diverse Data

Low Resource



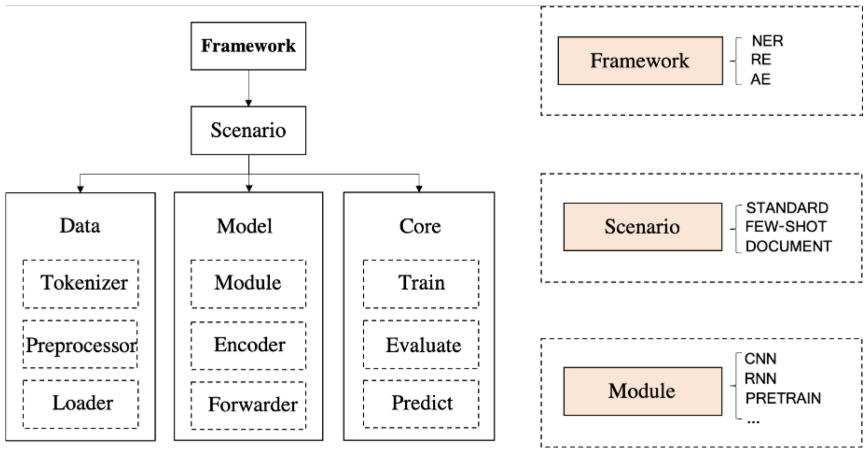
Off-the-shelf Usage

cnSchema
OpenKG 开放的中文知识图谱

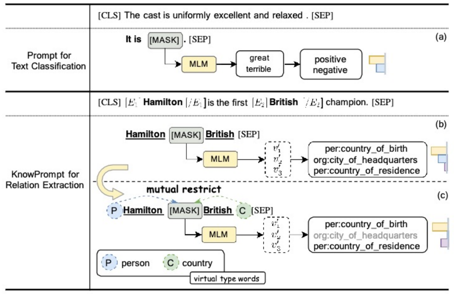
Flexible Training

DeepKE: with Traditional and Cutting-edge Methods

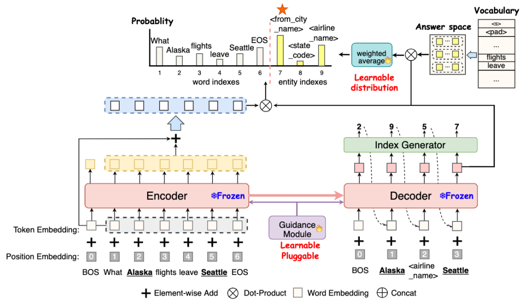
AACL IJCNLP 2022



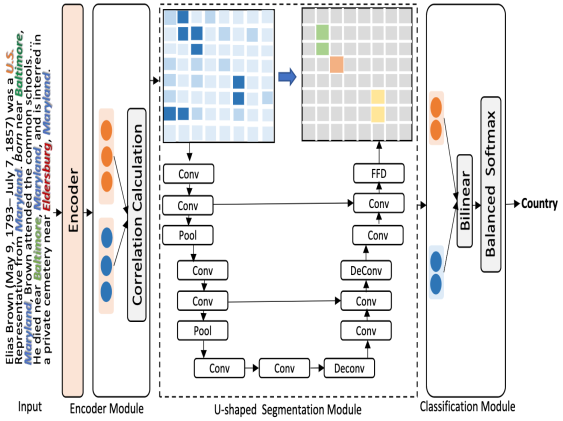

```python
# Import packages
from torch import optim
import deepke.relation_extraction.standard.models as models
from deepke.relation_extraction.standard.tools import preprocess, train, FocalLoss
# Preprocess data
preprocess(config)
# Choose the model, users can choose from CNN, RNN, Transformer, GCN, Capsule and BERT
__Model__ = {
        'cnn': models.PCNN,
        'rnn': models.BiLSTM,
        'transformer': models.Transformer,
        'gcn': models.GCN,
        'capsule': models.Capsule,
        'lm': models.LM,
    }
model = __Model__[config.model_name](config)
# Define the optimizer, scheduler and criterion
optimizer = optim.Adam(model.parameters(), lr=config.learning_rate, weight_decay=config.weight_decay)
scheduler = optim.lr_scheduler.ReduceLROnPlateau(optimizer, factor=config.lr_factor, patience=config.lr_patience)
criterion = FocalLoss()
# Train
train(epoch, model, train_dataloader, optimizer, criterion, device, writer, config)
# Save
model.save(epoch, config)
# Predict
model.load(config.saved_path, device=device)
model(preprocess(config.text))
```



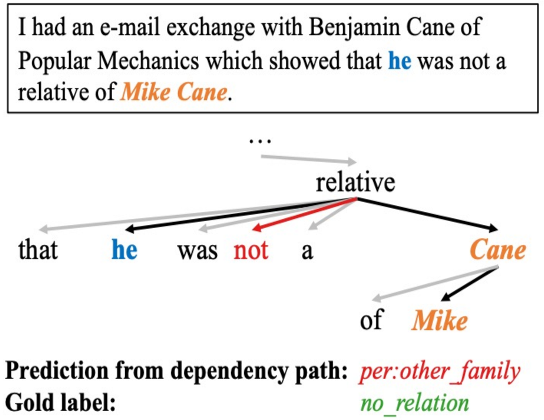
KnowPrompt WWW'22



LightNER COLING' 22



DocuNet IJCAI'21



C-GCN EMNLP'18

KnowPrompt: Knowledge-aware Prompt-tuning with Synergistic Optimization for Relation Extraction WWW'22
LightNER: A Lightweight Tuning Paradigm for Low-resource NER via Pluggable Prompting COLING'22
Document-level Relation Extraction as Semantic Segmentation IJCAI'21
Graph Convolution over Pruned Dependency Trees Improves Relation Extraction EMNLP'18

DeepKE: Usage

Step1 : go to the task folder

```
cd example/ner/standard
```

Step2: Train

```
python run.py
```

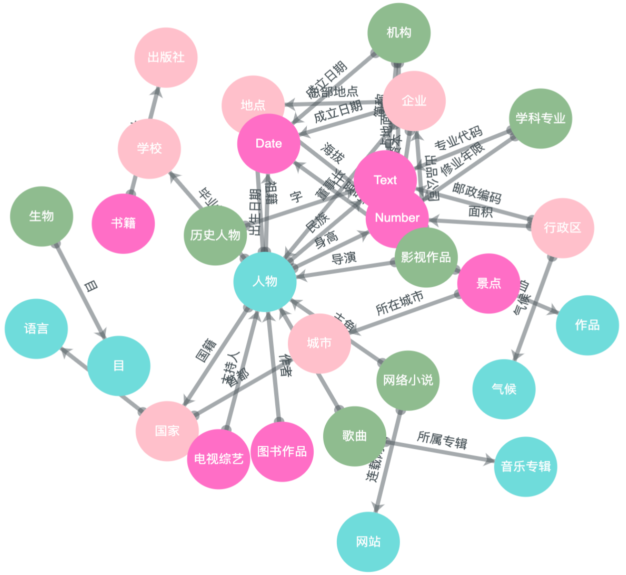
Step3 : Evaluation

```
python predict.py
```

Customize datasets,
auto-hyperparameter
tunning,**docker**

DeepKE: cnSchema-based Off-the-shelf Models

DeepKE provides off-the-shelf extraction models based on cnSchema: supporting **28 entity types** and **50 relation classes.**



20 Class + 50 Relation

Entity Recognition

DeepKE(NER), Roberta-wwm-ext, Chinese

Google dowload

Baidu Netdisk download(password:u022)

Relation Extraction

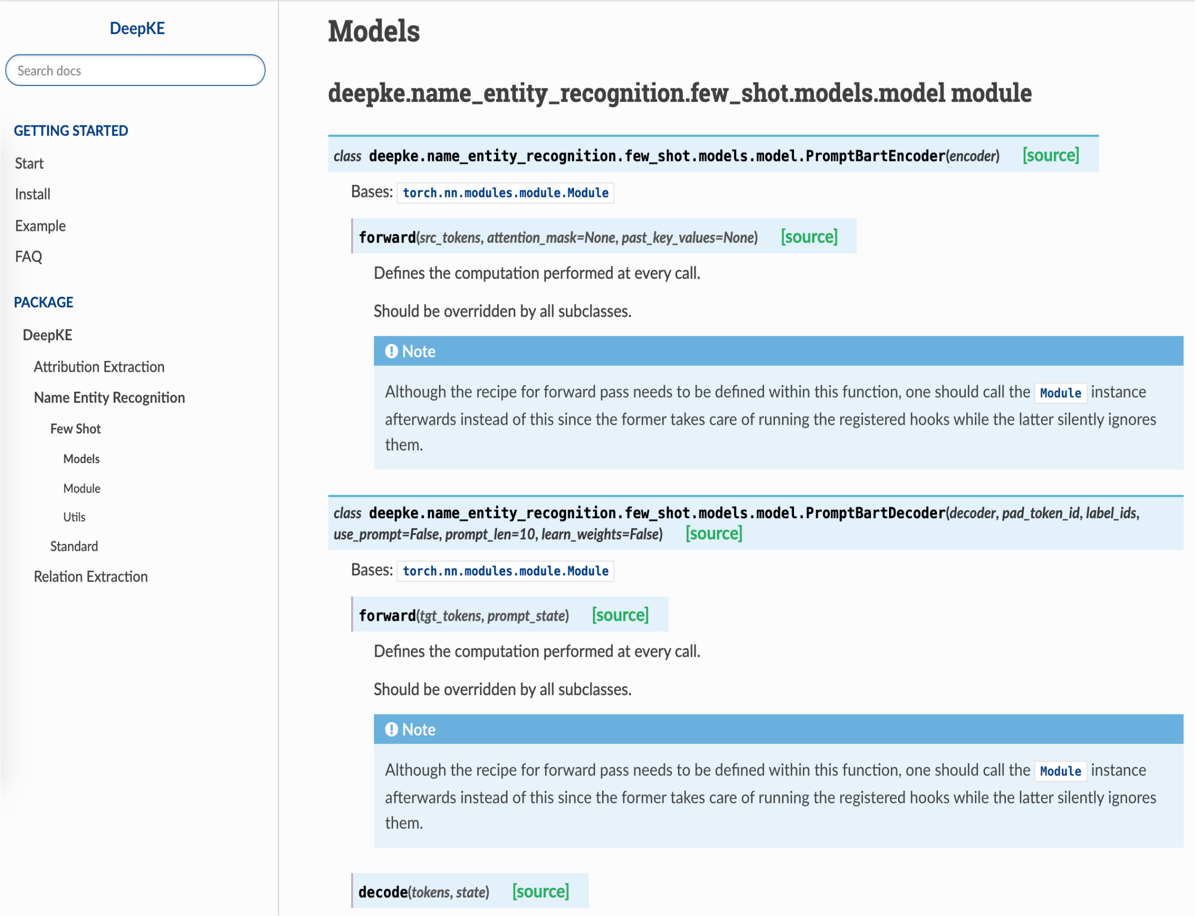
DeepKE(RE), Roberta-wwm-ext, Chinese

Google dowload

Baidu Netdisk download(password:29oe)

```
{
    "text": "查尔斯·阿兰基斯（Charles Aránguiz），1989年4月17日出生于智利圣地亚哥，智利职业足球运员",
    "spo_list": [{
        "predicate": "出生地",
        "object_type": "地点",
        "subject_type": "人物",
        "object": "圣地亚哥",
        "subject": "查尔斯·阿兰基斯"
    }, {
        "predicate": "出生日期",
        "object_type": "Date",
        "subject_type": "人物",
        "object": "1989年4月17日",
        "subject": "查尔斯·阿兰基斯"
    }]
}
```

DeepKE: Documents&Tutorials



DeepKE

Search docs

GETTING STARTED

Start
Install
Example
FAQ

PACKAGE

DeepKE
Attribution Extraction
Name Entity Recognition
Few Shot
Models
Module
Utils
Standard
Relation Extraction

Models

deepke.name_entity_recognition.few_shot.models.model module

class `deepke.name_entity_recognition.few_shot.models.model.PromptBartEncoder`(*encoder*) [source]

Bases: `torch.nn.modules.module.Module`

`forward`(*src_tokens, attention_mask=None, past_key_values=None*) [source]

Defines the computation performed at every call.

Should be overridden by all subclasses.

Note

Although the recipe for forward pass needs to be defined within this function, one should call the `Module` instance afterwards instead of this since the former takes care of running the registered hooks while the latter silently ignores them.

class `deepke.name_entity_recognition.few_shot.models.model.PromptBartDecoder`(*decoder, pad_token_id, label_ids, use_prompt=False, prompt_len=10, learn_weights=False*) [source]

Bases: `torch.nn.modules.module.Module`

`forward`(*tgt_tokens, prompt_state*) [source]

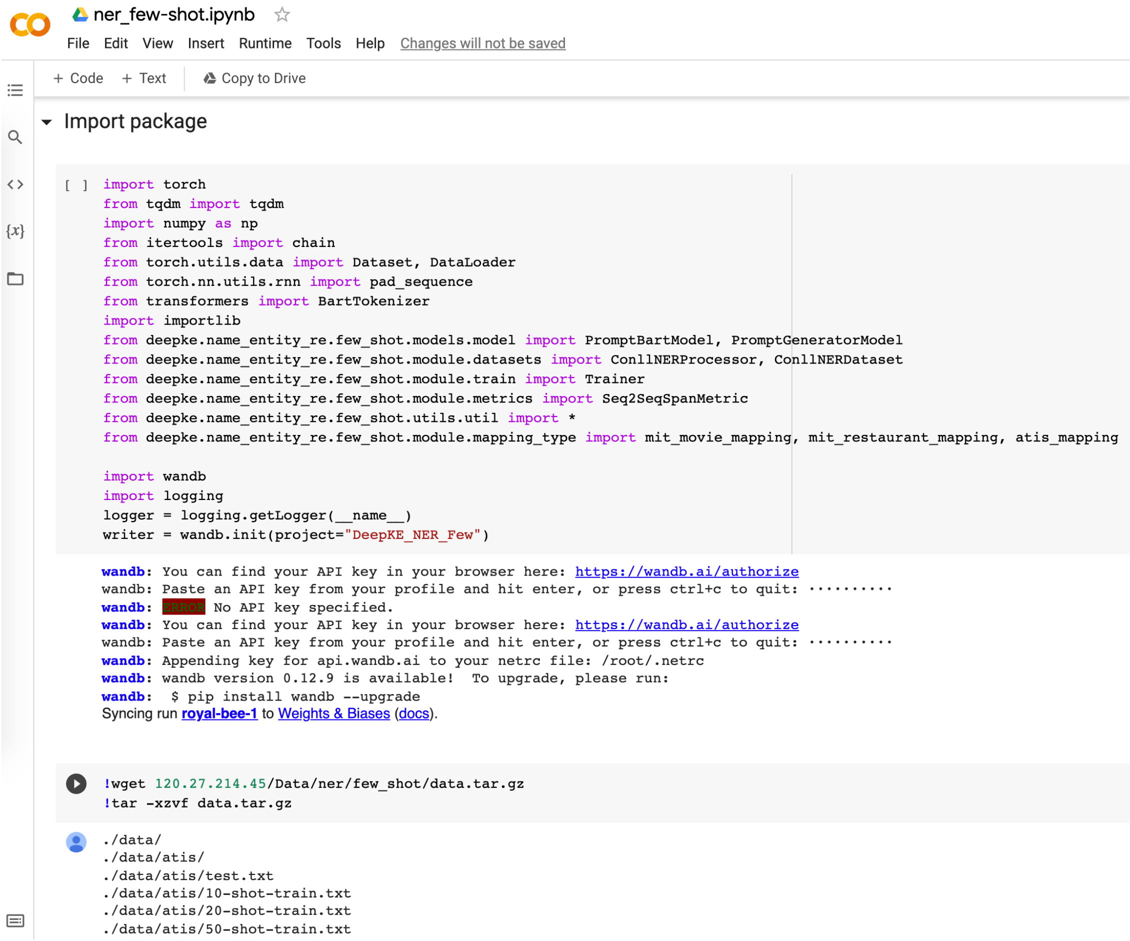
Defines the computation performed at every call.

Should be overridden by all subclasses.

Note

Although the recipe for forward pass needs to be defined within this function, one should call the `Module` instance afterwards instead of this since the former takes care of running the registered hooks while the latter silently ignores them.

`decode`(*tokens, state*) [source]



ner_few-shot.ipynb

File Edit View Insert Runtime Tools Help Changes will not be saved

+ Code + Text Copy to Drive

Import package

```
import torch
from tqdm import tqdm
import numpy as np
from itertools import chain
from torch.utils.data import Dataset, DataLoader
from torch.nn.utils.rnn import pad_sequence
from transformers import BartTokenizer
import importlib
from deepke.name_entity_re.few_shot.models.model import PromptBartModel, PromptGeneratorModel
from deepke.name_entity_re.few_shot.module.datasets import Conll1NERProcessor, Conll1NERDataset
from deepke.name_entity_re.few_shot.module.train import Trainer
from deepke.name_entity_re.few_shot.module.metrics import Seq2SeqSpanMetric
from deepke.name_entity_re.few_shot.utils.util import *
from deepke.name_entity_re.few_shot.module.mapping_type import mit_movie_mapping, mit_restaurant_mapping, atis_mapping

import wandb
import logging
logger = logging.getLogger(__name__)
writer = wandb.init(project="DeepKE_NER_Few")
```

```
wandb: You can find your API key in your browser here: https://wandb.ai/authorize
wandb: Paste an API key from your profile and hit enter, or press ctrl+c to quit: ··········
wandb: ERROR No API key specified.
wandb: You can find your API key in your browser here: https://wandb.ai/authorize
wandb: Paste an API key from your profile and hit enter, or press ctrl+c to quit: ··········
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
wandb: wandb version 0.12.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade
Syncing run royal-bee-1 to Weights & Biases (docs).
```

```
!wget 120.27.214.45/Data/ner/few_shot/data.tar.gz
!tar -xzvf data.tar.gz
```

```
./data/
./data/atis/
./data/atis/test.txt
./data/atis/10-shot-train.txt
./data/atis/20-shot-train.txt
./data/atis/50-shot-train.txt
```

Comprehensive documents

Google Colab Tutorials

The End

Github: https://github.com/NLP-Tutorials/AACL-IJCNLP2022-KGC-Tutorial

Questions: zhangningyu@zju.edu.cn, tgui@fudan.edu.cn, nanguo2021@bupt.edu.cn