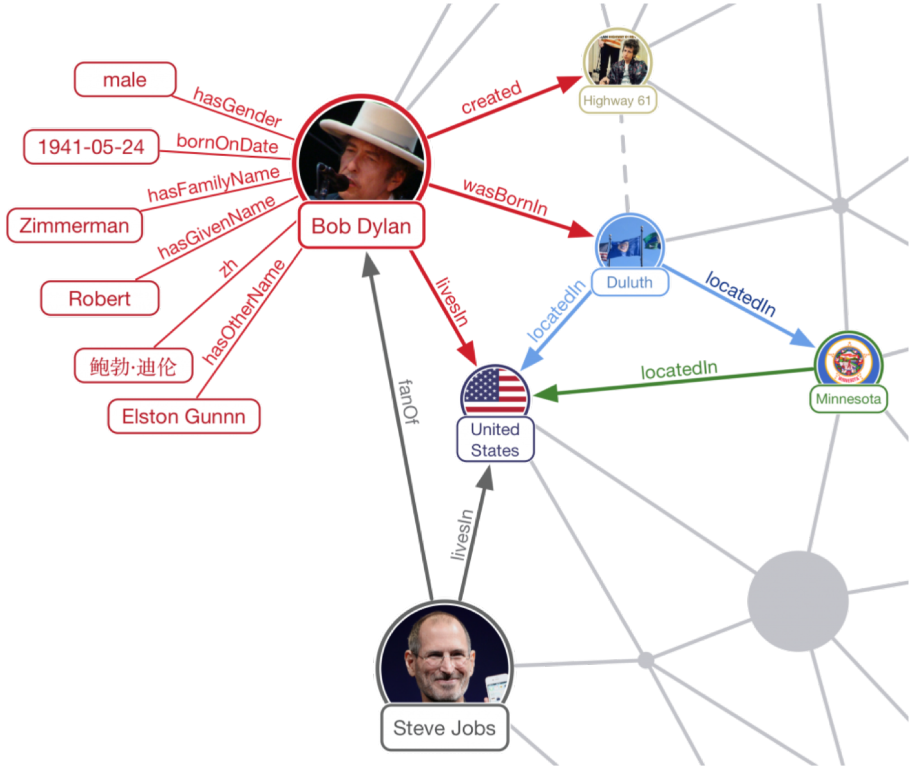
Knowledge Graph Question Answering by Pre-training Models

Jing Zhang

Renmin University of China

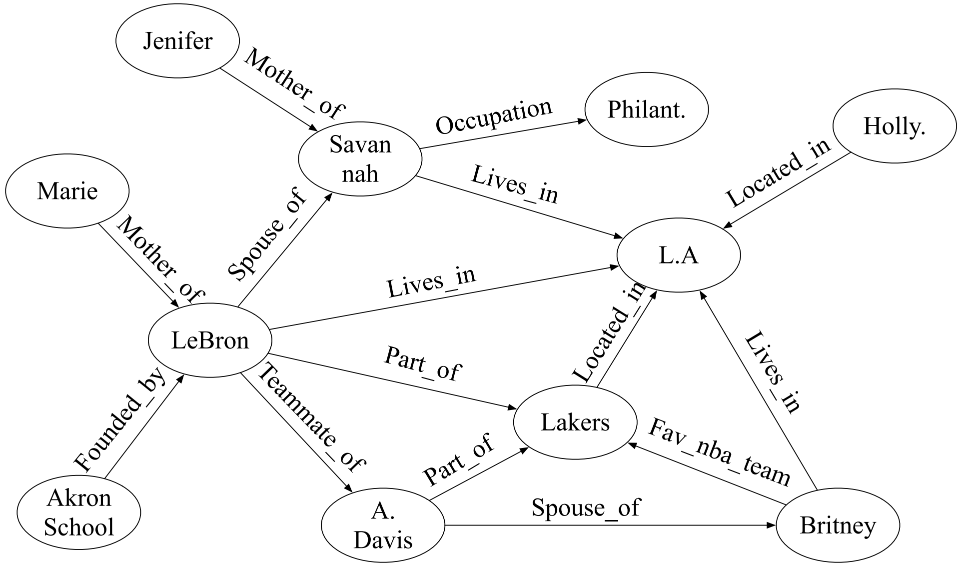
Knowledge Graph

➢A set of facts represented as triplets (head entity, relation, tail entity)



Knowledge Graph Question Answering

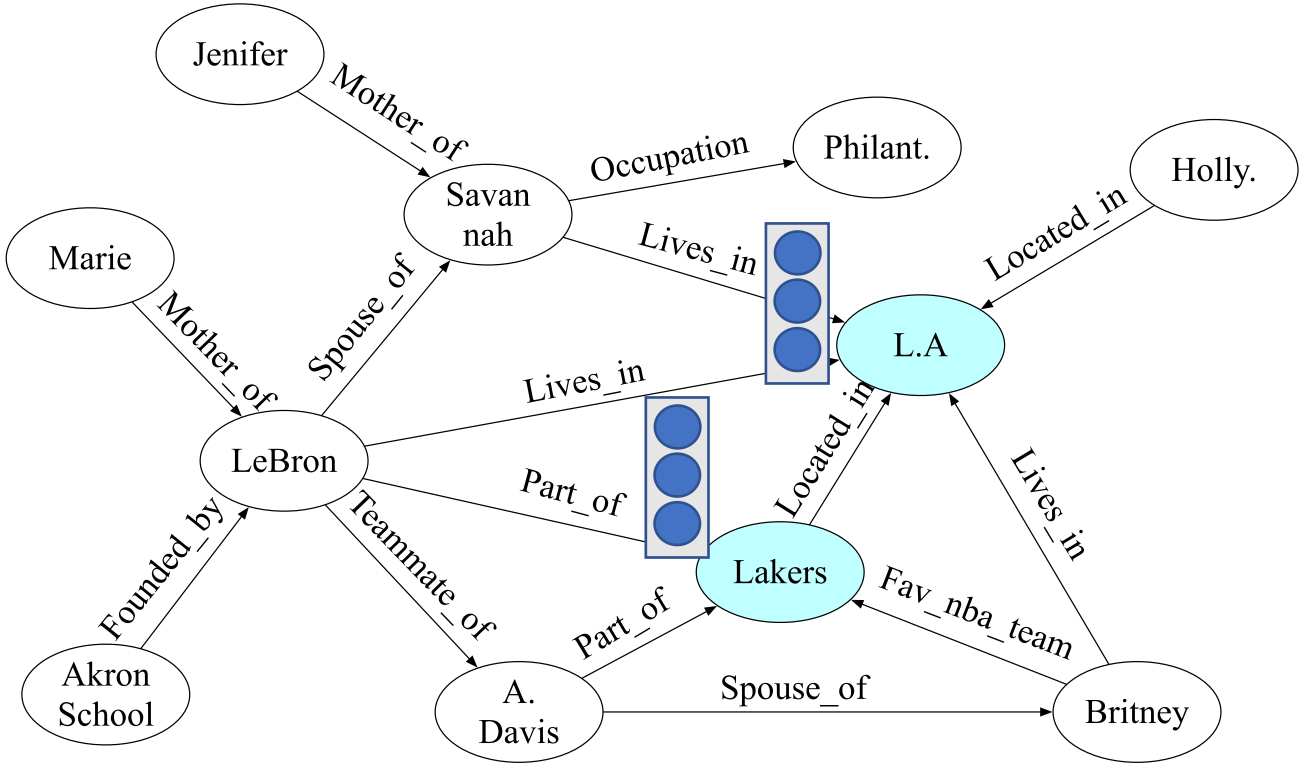
- Deduce entities on KGs as the answers to the given query.
- A query is a textural question (knowledge graph question answering, KGQA).

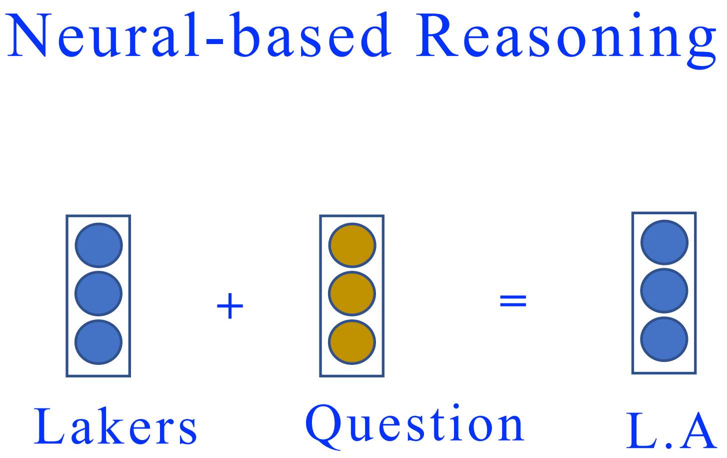


Textual question: Where do the spouses of the teammates of Lakers usually live?

Answer: L.A

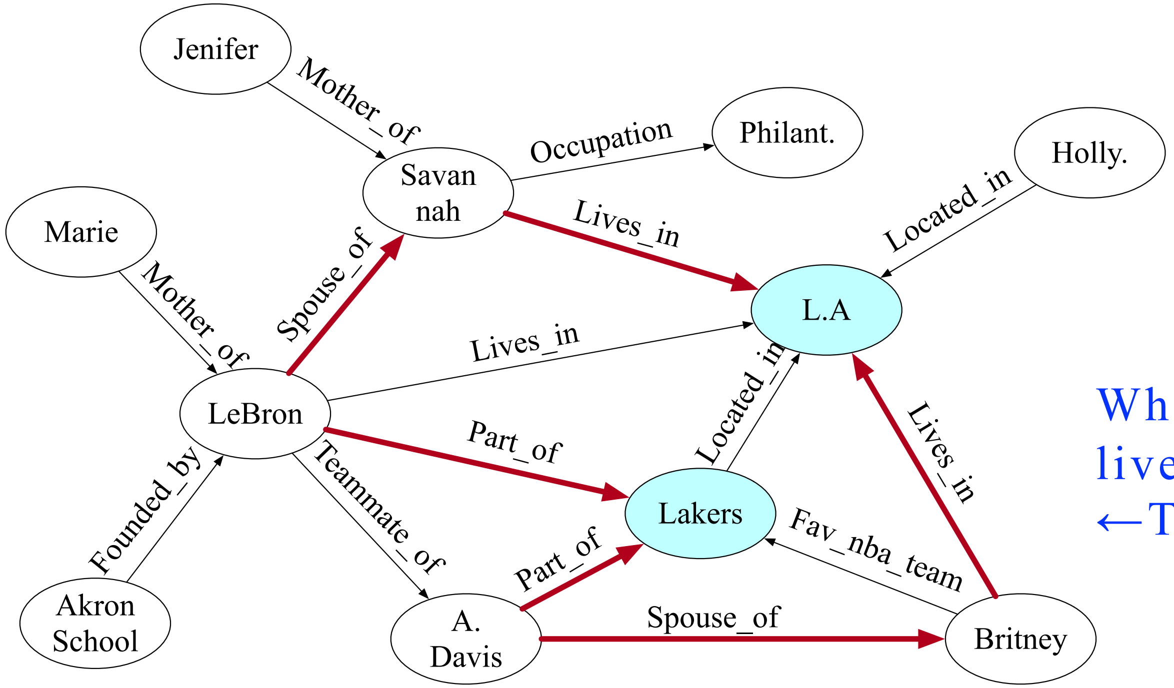
Neural-based KGQA Reasoning





Textual question: Where do the spouses of the teammates of Lakers usually live?
Reasoning result: L.A

Symbolic-based KGQA Reasoning



Symbolic-based Reasoning

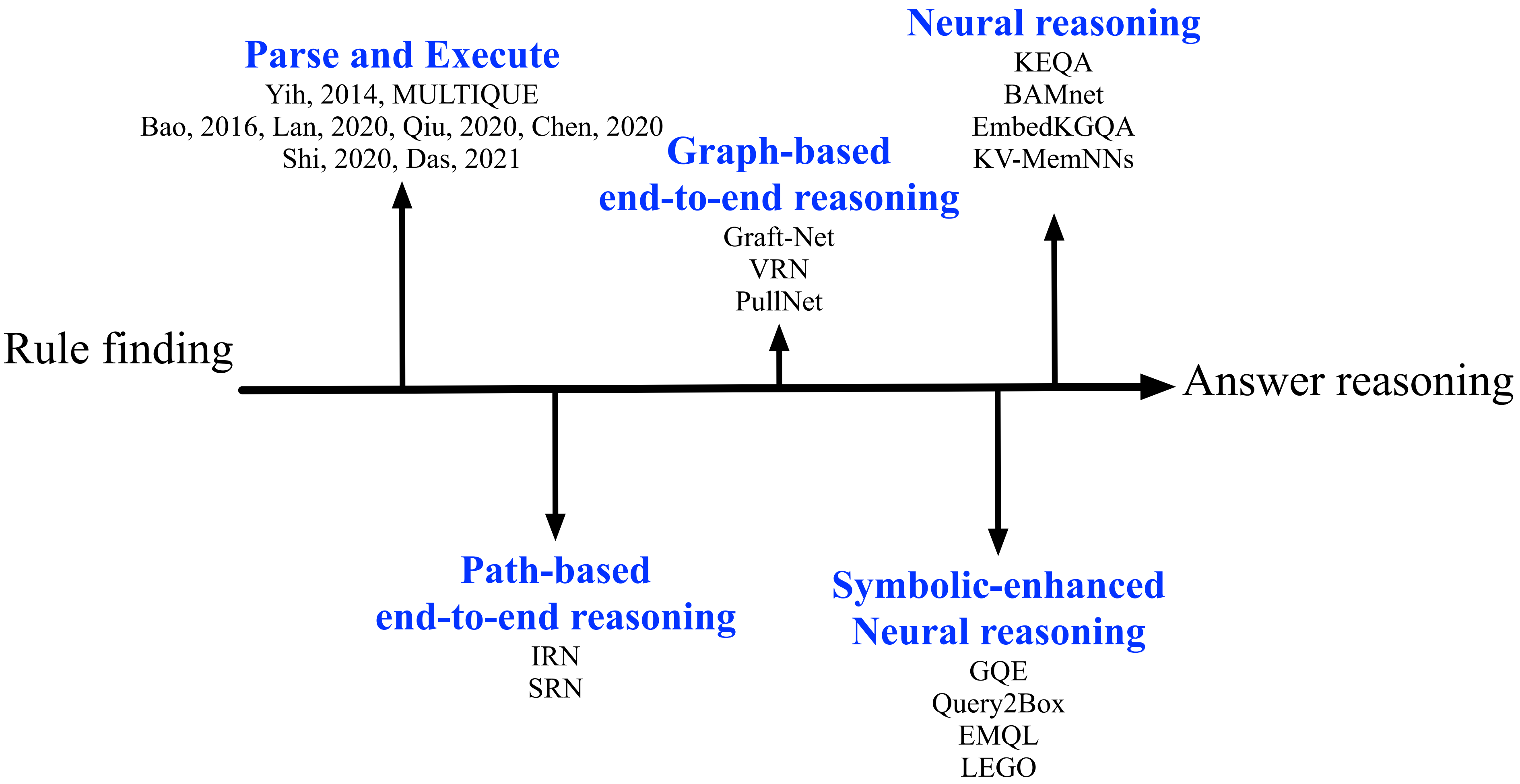
Where do the spouses of the teammates of x usually live?

$\leftarrow$Teammate_of(x,z) $\wedge$ Spouse_of(z,w) $\wedge$ Lives_in (w,y)

Textual question: Where do the spouses of the teammates of Lakers usually live?

Reasoning result: L.A

Neural-Symbolic based KBQA Reasoning



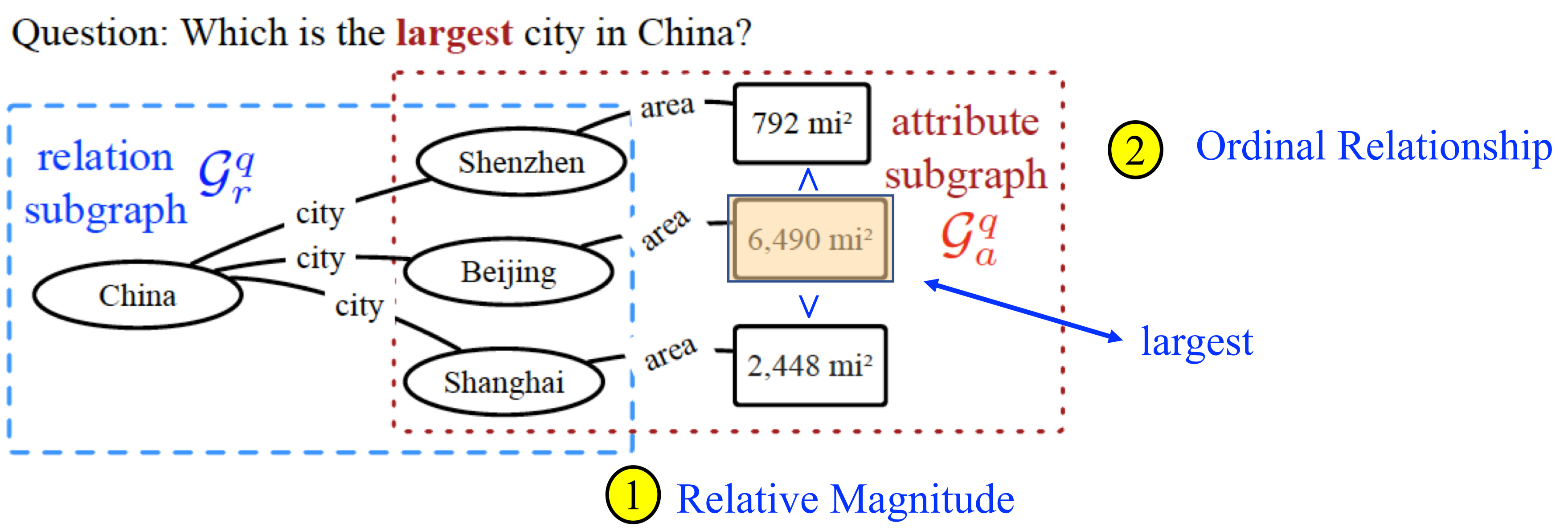
Remaining Challenges of KBQA

- Complex Questions
  - Symbolic reasoning relies heavily on large annotated question-logic form pairs.
  - Neural reasoning is not easy to address various constraints.

- Evidence is Inadequate
  - Symbolic reasoning highly probably fails.
  - Neural reasoning on incomplete KGs is also inadequate.

Ordinal Constrained Complex Question

➢The answers of such question should be selected from a ranked set based on ordinal determiners in the question as ranking criteria.

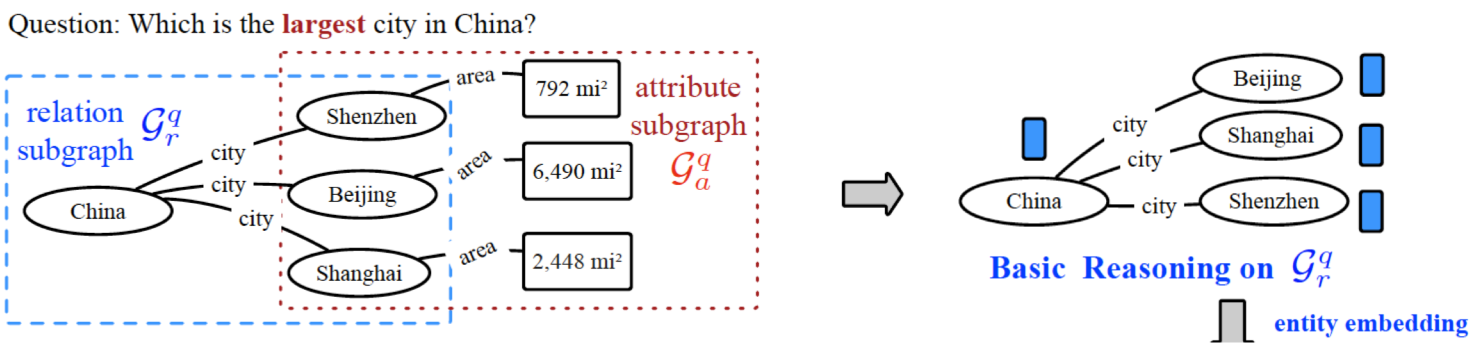


A Pretraining Numerical Reasoning Model for Ordinal Constrained Question Answering on Knowledge Base. EMNLP 2021

Proposed Method

➢ We answer an ordinal constraint question by reasoning in two steps.

➢ Basic reasoning: infers entity embeddings that can encode the semantic relationships between entities and the question.

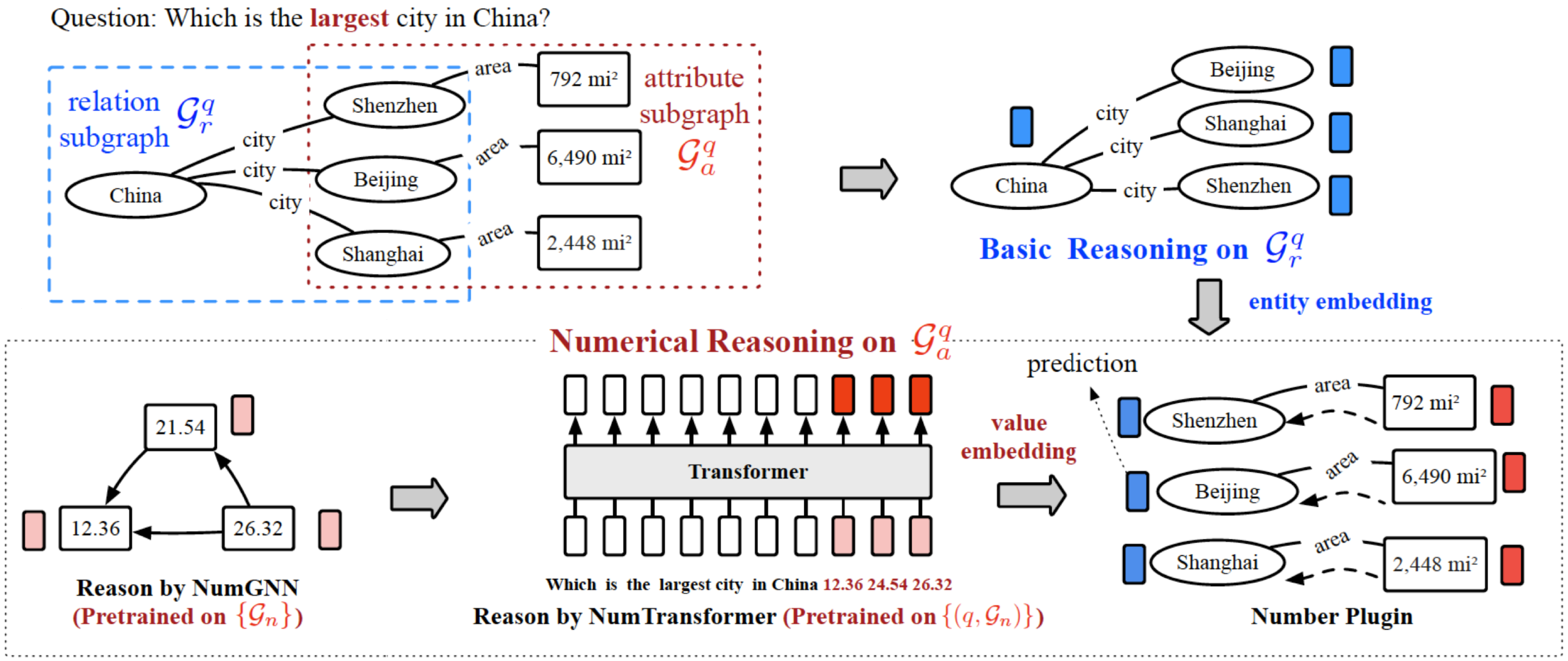


Proposed Method

➢ A pretraining numerical reasoning model consisting of NumGNN and NumTransformer, guided by explicit self-supervision signals.

Ordinal Constraint

Magnitude



Pretraining

- NumGNN: preserve the relative magnitude of numbers 1≺2≺3

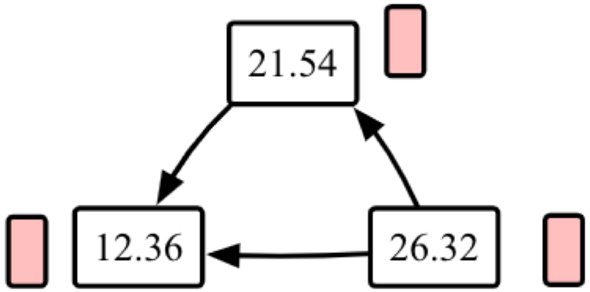
- **Number Graph Construction:**
  - Numbers from the same relations.
  - The "greater" edge

- **Number Representation:**

$$\tilde{\mathbf{v}}_i^{(l-1)} = \frac{1}{|\mathcal{N}_n(i)|}\Big(\sum_{v_j \in \mathcal{N}_n(i)} \alpha_j \mathrm{MLP}(\mathbf{v}_j^{(l-1)})\Big)$$

- **Number-aware triplet ranking loss:**
  - The relative distance between numbers.

$$\ell = \sum_{(v_s, v_m, v_b) \in \mathcal{T}} \max(0, \epsilon + g(\mathbf{v}_s, \mathbf{v}_m) - g(\mathbf{v}_s, \mathbf{v}_b))$$



Reason by NumGNN
(Pretrained on $\{\mathcal{G}_n\}$)

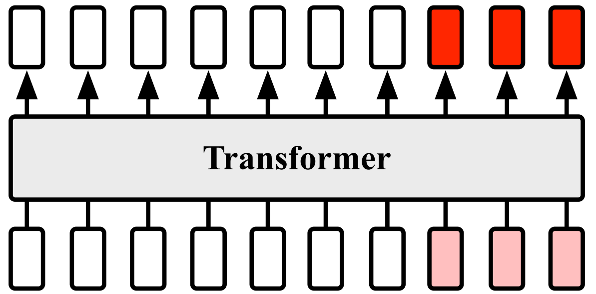
Pretraining

- NumTransformer: learn the ordinal properties of numbers
  - make the embedding of 1 in 1<2<3 closer to ``smallest'' than 2 and 3

- **Question-aware Number Graph Construction:**
  - Ordinal constrained question and
  - Numbers from the most relevant relations.

- **Number Representation:**
  - NumGNN on numbers
  - $\{\mathbf{v}^{(L')}\} = \text{Transformer}([\mathbf{h}_q^{(0)}; \{\mathbf{v}^{(L)}\}])$

- **Cross-entropy loss:**
  - predict the ground truth number based on its output embedding.



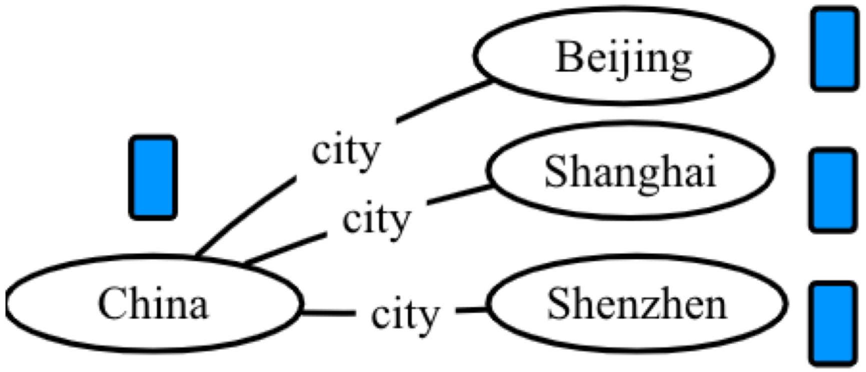
Reason by NumTransformer (Pretrained on $\{(q, \mathcal{G}_n)\}$)

Training

- Basic reasoning

- Infer entity embeddings that can encode the semantic relationships between entities and the question.

Question: Which is the **largest** city in China?



Basic Reasoning on $\mathcal{G}_r^q$

Training

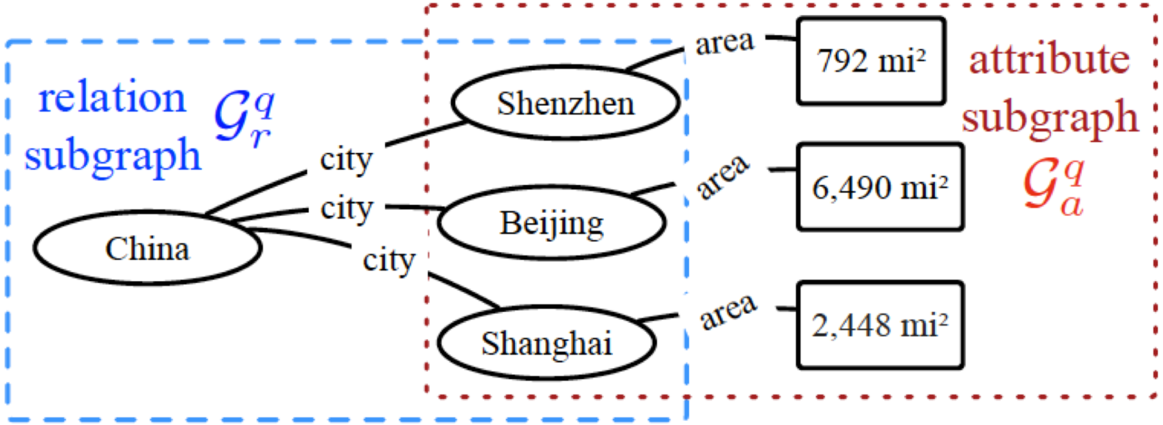
- Numerical reasoning

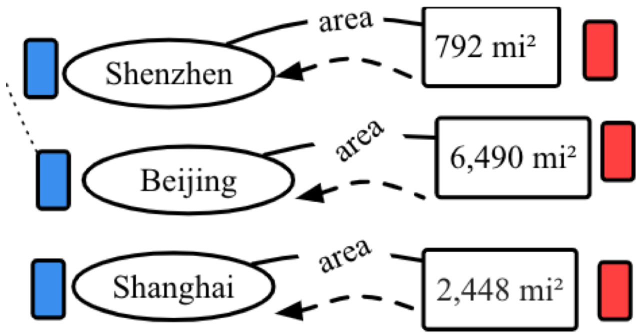
- Attribute Subgraph Retrieval:
  - Extract the numerical attribute triplets to compose the attribute subgraph

- Number Embedding Inference:
  - NumGNN & NumTransformer

- Number Embedding Plugin:

Question: Which is the **largest** city in China?





Number Plugin

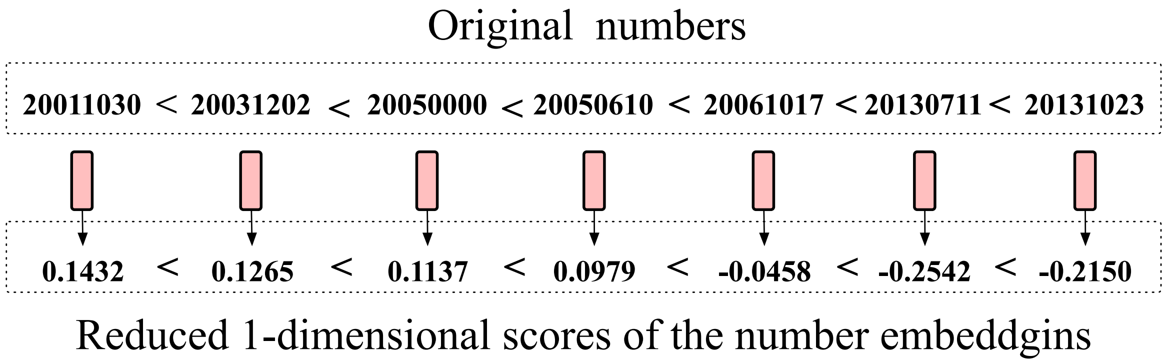
Experiments

Model	WebQSP All	WebQSP Ordinal	CWQ All	CWQ Ordinal
GRAFT-Net	66.4	28.4	36.8	19.3
GRAFT-Net+Num	**67.4**	**43.2**	**37.3**	**25.9**
EmbedKGQA	46.0	35.4	32.0	20.0
EmbedKGQA+Num	**47.6**	**45.4**	32.0	**22.4**
NSM	68.5	33.3	46.3	24.4
NSM+Num	**68.6**	**38.5**	**47.4**	**28.4**

GRAFT-Net, EmbedKGQA and NSM perform better on numerical reasoning when attached with the pretrained numerical module.

Experiments

	WebQSP All	WebQSP Ordinal	CWQ All	CWQ Ordinal
GRAFT-Net	66.4	28.4	36.8	19.3
+ NumGNN	66.4	32.7	36.9	21.6
+ NumGNN (Pre-trained)	66.5	37.8	36.9	22.3
+ Num	66.4	33.7	36.8	20.8
+ Num (Pre-trained)	**67.4**	**43.2**	**37.3**	**25.9**

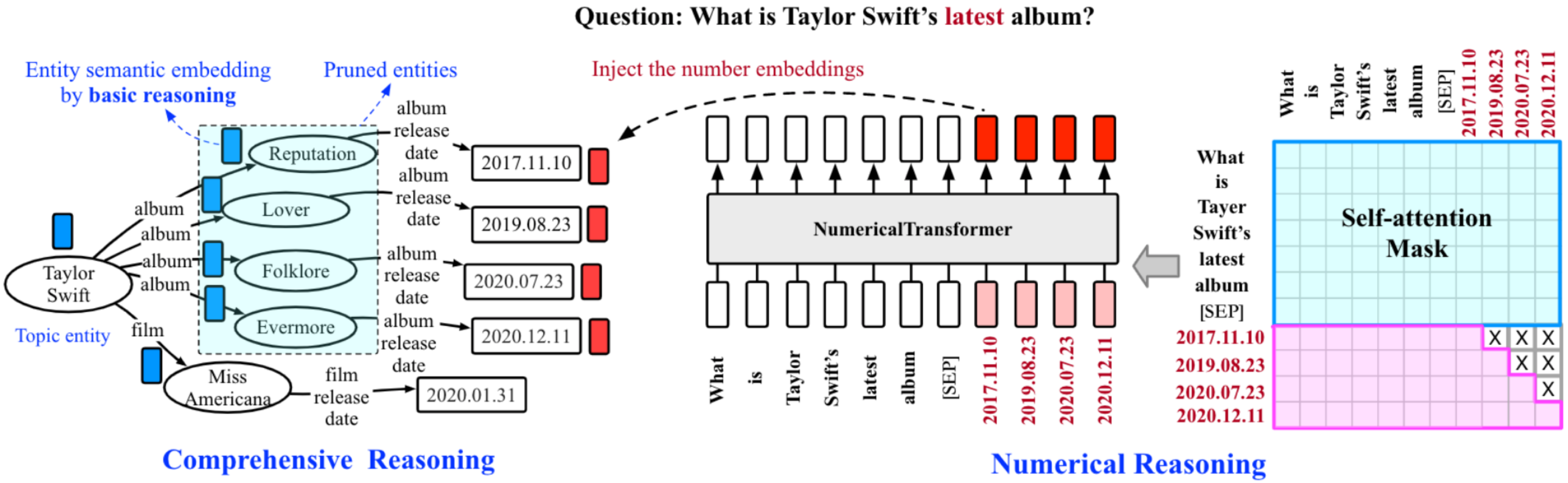


- Both NumGNN and NumTransformer takes positive effect.
- The pretraining loss function provides positive guidance for NumGNN and NumTransformer.

- The reduced 1-dimensional scores of numbers can still preserve the relative magnitude.

Improvement

- NumGNN + NumTransformer => NumTransformer
- Comprehensive reasoning: prune entities by basic reasoning



Two Pretraining Tasks

Two generated datasets:

	Ordinal determiner / Question q	Relation r	Numbers $\mathcal{V}_r$			answer v_q
QIND:	latest	music.album.release_date	1994.06.18	1997.02.28	2002.11.06	2003.11.22
QGND:	Of the films in which Kim Richards acted, which was released latest?	film.film.release_date	2000.10.11	2004.07.06	2008.05.06	2009.08.04

Objectives:

- Number-aware triplet ranking loss:

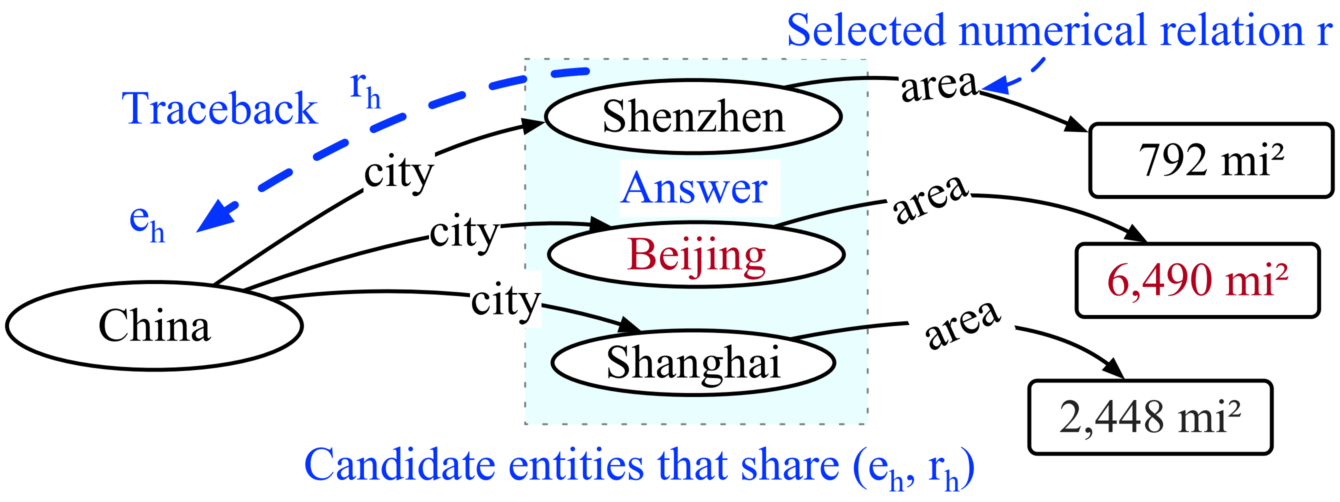
reflect the relative distance between numbers rather than the absolute magnitude.

- Cross-entropy loss:

predict the ground truth number based on its output embedding.

QA Pair Augmentation

- Generate ordinal constrained QA pairs.



Template: What is the [r_h] of [e_h] that has the [ordinal determiner] [r] ?

Instance: What is the city of China that has the largest area ?

Experiments

Model	WebQSP All	WebQSP Ordinal	CWQ All	CWQ Ordinal
SP-based models				
SPARQA	-	-	31.6	6.3
QGG	**73.0**	**61.2**	36.9	24.9
Embedding-based models				
KV-Mem	46.6	33.3	18.4	11.7
BAMnet	55.6	41.0	-	-
EmbedKGQA	46.0	35.4	32.0	20.0
GRAFT-Net	66.4	28.4	36.8	19.3
NT-GRAFT-Net (**Ours**)	67.2	41.0	37.3	25.8
NSM	68.5	33.3	46.3	24.4
NT-NSM (Ours)	69.1	46.2	**48.9**	**42.1**

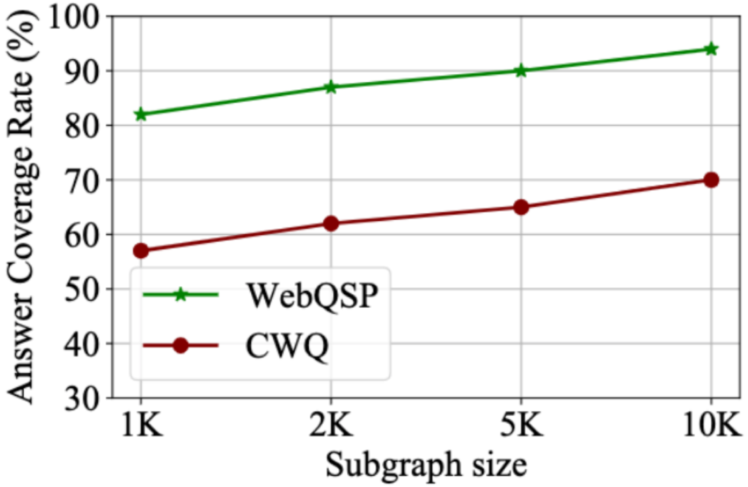
- The simple NumTransformer is able to address ordinal constrained KBQA well.
- The framework can be adapted to other embedding-based KBQA models.

Ablation Studies				
NT-NSM w/o SAM	68.2	38.5	47.8	35.5
NT-NSM w/o SNE	68.7	43.6	48.3	40.6
NT-NSM w/o QGND	68.7	41.0	48.3	41.1
NT-NSM w/o QIND	68.1	38.5	47.7	36.0
NT-NSM w/o NPL	68.4	41.0	48.3	37.6
NT-NSM w/o NTL	68.8	43.6	48.4	41.6
NT-NSM w/o Pre-train	68.6	38.5	47.0	28.9
NT-NSM w/o DA	68.6	41.0	47.2	30.9

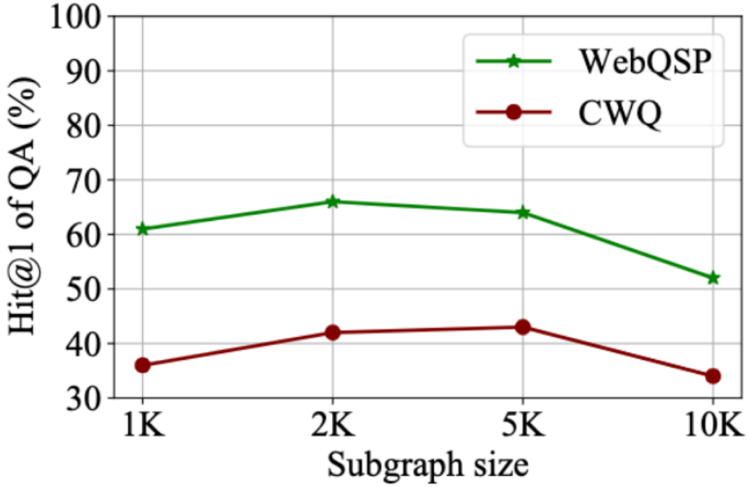
- Pre-training on the automatically generated training data provides the model with numerical reasoning skills.
- Adding pseudo ordinal constrained QA data improves the QA performance.

Inadequate Evidence

- Subgraph retrieval is crucial to the overall QA performance
  - A small subgraph is highly likely to exclude the answer.
  - A large one might introduce noises.



(a) Answer Coverage Rate



(b) Hits@1 of QA

Subgraph Retriever + Answer Reasoner

$$p(a|G,q) = \sum_{\mathcal{G}} \underbrace{p_\phi(a|q,\mathcal{G})} \underbrace{p_\theta(\mathcal{G}|q)}.$$

Retriever:

Reasoner: The likelihood of the answer a given q and $\mathcal{G}$

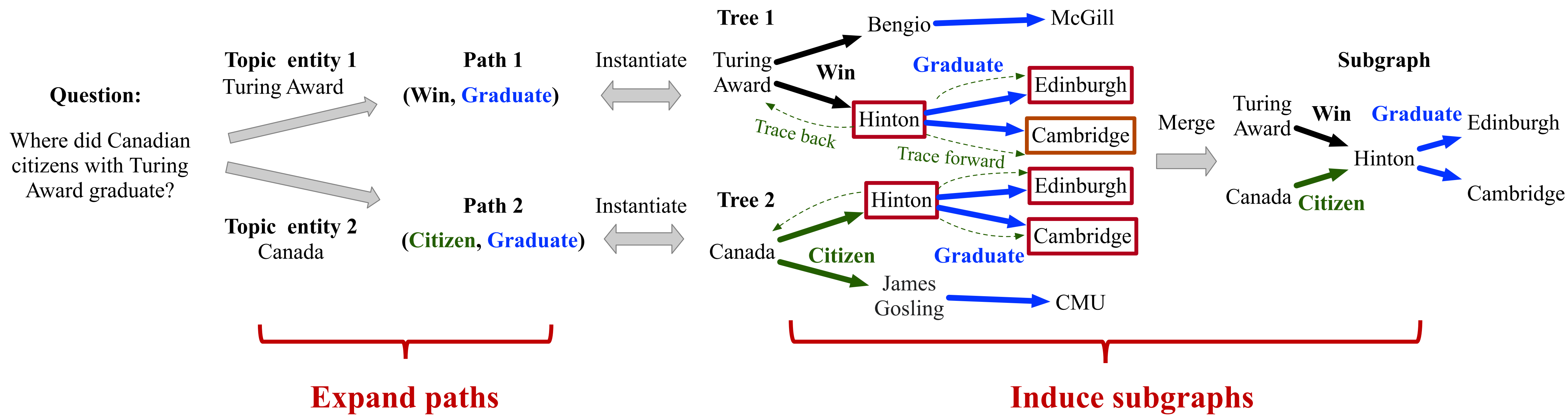
Prior distribution over a latent subgraph $\mathcal{G}$ conditioned on a question q

Via drawing a sample $\mathcal{G}$, the likelihood of training data can be approximated as:

$$\mathcal{L}(\theta,\phi) = \max_{\theta,\phi} \sum_{(q,a,\mathcal{G})\in\mathcal{D}} \log p_\phi(a|q,\mathcal{G}) + \log p_\theta(\mathcal{G}|q),$$

The retriever is decoupled from the reasoner by firstly training the retriever and then the reasoner on the sampled subgraphs by the retriever.

Subgraph Retriever



Dual encoder: $s(q, r) = f(q)^{\top} h(r)$

Merge the same entities in different subgraph.

Question update: $f(q^{(t)}) = \text{RoBERTa}([q; r_1; \cdots; r_t])$

Path end: $$p(r|q^{(t)}) = \frac{1}{1 + \exp\left(s(q^{(t)}, r) - s(q^{(t)}, \text{END})\right)}$$

Pre-training Subgraph Retriever

Weakly Supervised Pre-training

We retrieve the shortest paths to each answer as the supervision signals (q, a, p).

Unsupervised Pre-training

We leverage NYT, a distant supervision dataset for relation extraction to construct pseudo (q, a, p) labels.

$$(s_1, (\boxed{e_1}, r_1, e_2)) \qquad (s_2, (e_2, r_2, \boxed{e_3}))$$

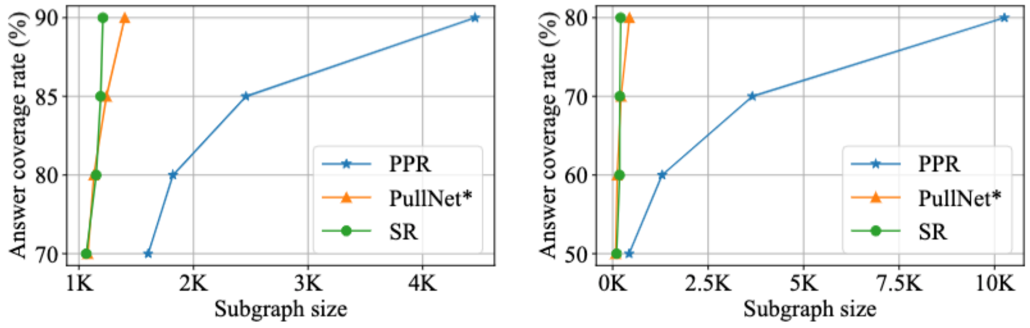
Topic entity

Answer

Question: $s_1 + s_2$

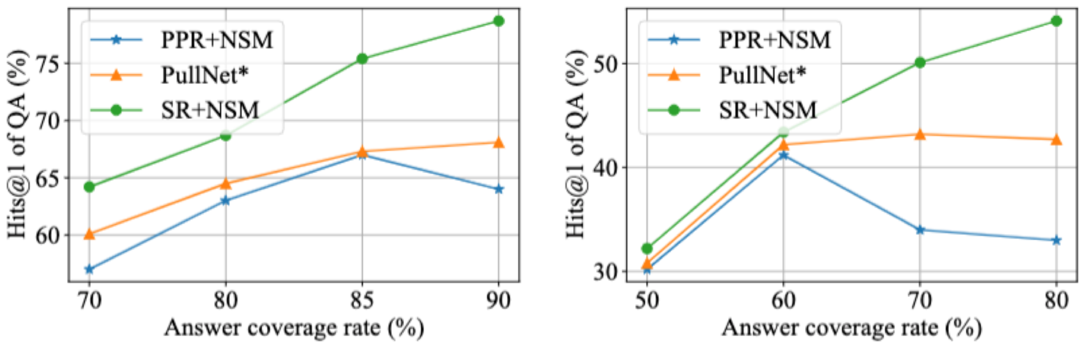
Path: $r_1 + r_2$

Retriever Evaluation



(a) WebQSP (b) CWQ

Comparison of the answer coverage rate under various subgraph sizes.

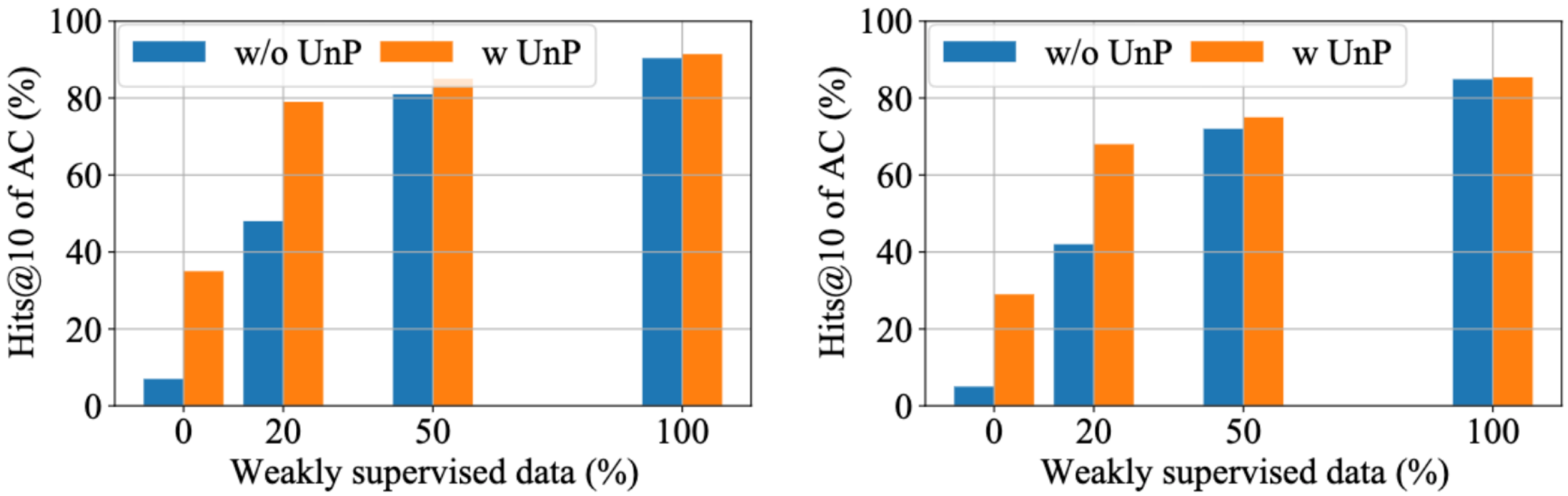


(c) WebQSP (d) CWQ

Comparison of the QA performance under various answer coverage rates.

The proposed subgraph retriever can obtain smaller but higher-quality subgraphs.

Pre-training Evaluation



(a) WebQSP (b) CWQ

Adding only 20% weakly supervised data after unsupervised pre-training can achieve comparable performance with 100% weakly supervised data.

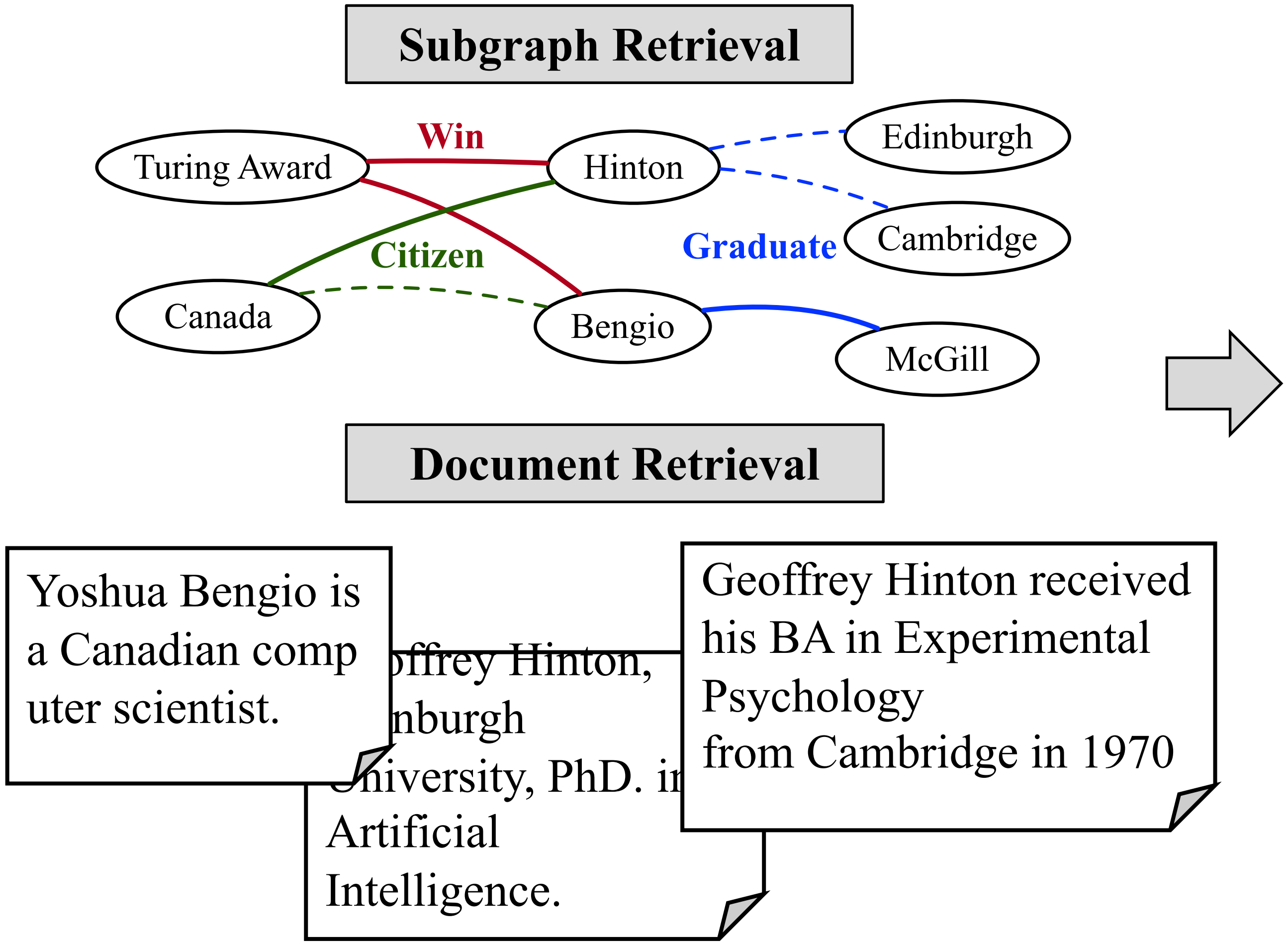
Overall QA Evaluation

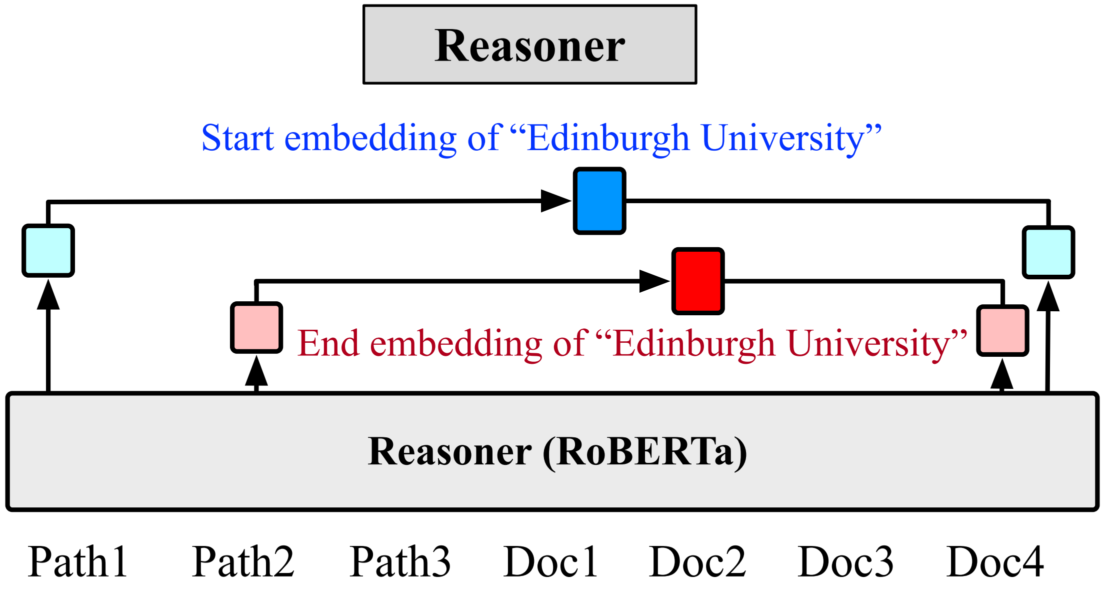
Model	WebQSP		CWQ	
	Hits@1	F1	Hits@1	F1
KV-Mem	46.6	34.5	18.4	15.7
EmbedKGQA	46.0	-	32.0	-
BAMnet	55.6	51.8	-	-
GRAFT-Net (GN)	66.4	60.4	36.8	32.7
NSM	68.5	62.8	46.3	42.4
PullNet	68.1	-	45.9	-
SR+NSM	82.7	74.1	57.6	48.3
SR+GN	80.3	73.9	55.1	47.8
SR+NSM w E2E	**83.2**	**74.5**	**58.4**	**50.2**
SR+GN w E2E	81.9	74.1	55.7	48.7

The overall evaluation shows that the proposed subgraph retriever (SR) takes effect in improving the QA performance when injecting it before a subgraph-oriented reasoner.

Document Evidence

Question: Where did Canadian citizens with Turing Award graduate?



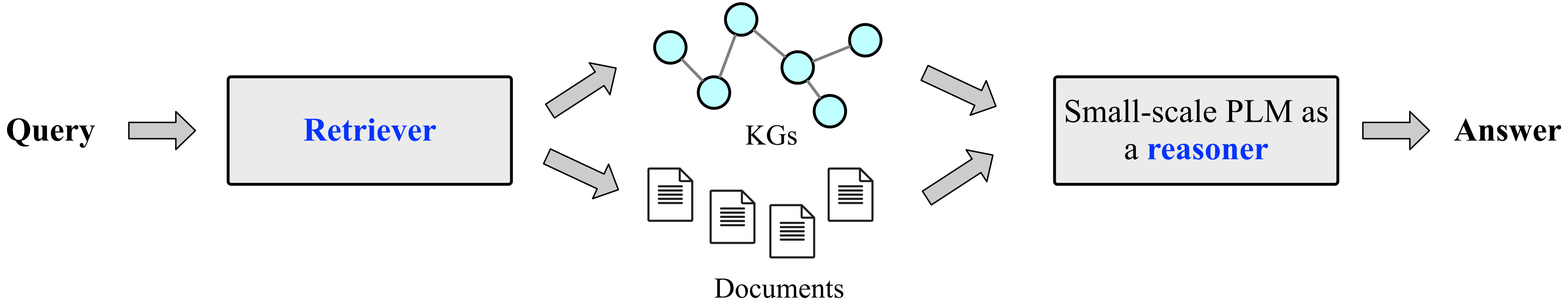


Results: WebQSP dataset: 50% KG + Wikipages

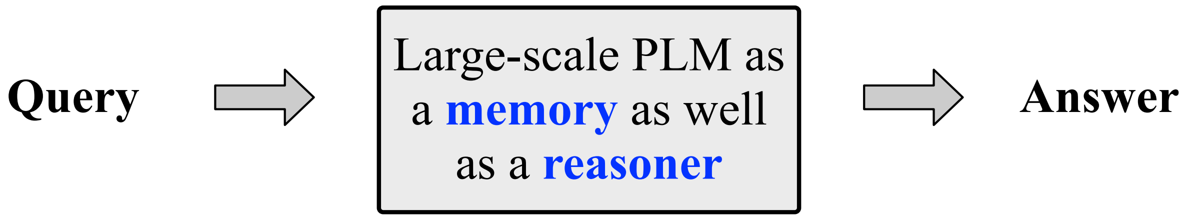
PullNet: 51.9% Hits@1
Ours: 63.6% Hits@1

Future Work on Large-scale PLM

➢Existing work usually treats the small-scale PLM as a reasoner.

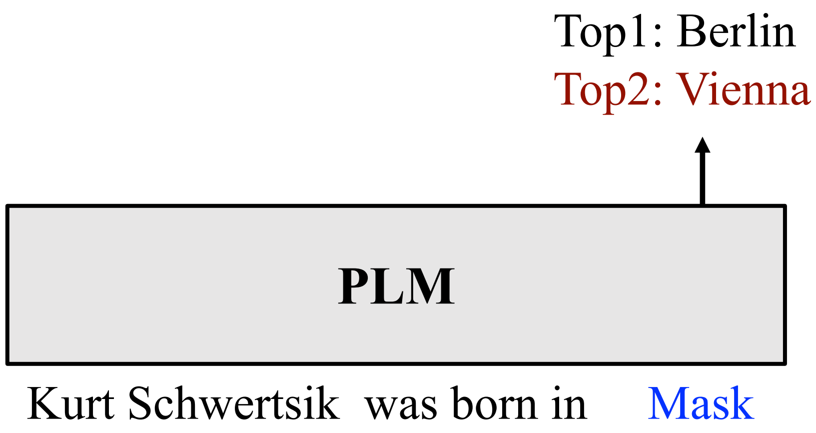


➢Can we treat the large-scale PLM as a memory in addition to a reasoner?



Future Work on Large-scale PLM

- Possible solution for knowledge probing: prompt tuning.
- Does PLM know an entity?



Berlin and Vienna are both German speaking area

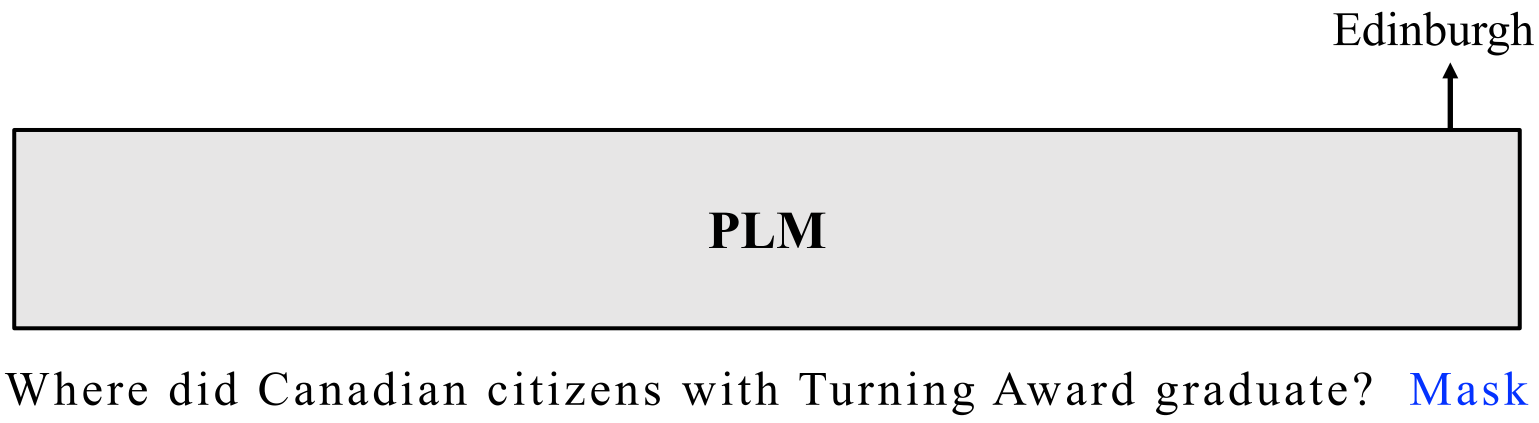
PLM does not know "Kurt Schwertsik"

The answer is predicted according to the token of the name

33% Hits@1

LAMA: the dataset for knowledge probing

Future Work on Large-scale PLM



- What is the prompt if the query is a textual question?
  - Entity name? Entity concept? Relation name? Question type?
- If firstly generating the context, and then predicting the answer, how to make sure the context is relevant to the question?

Thank You