

New Frontiers of Knowledge Graph Reasoning: Recent Advances and Future Trends



Lihui Liu

lihuil2@illinois.edu



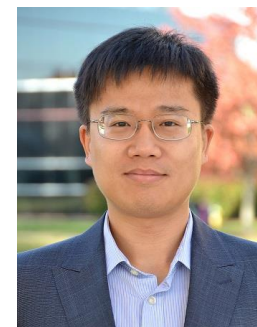
Zihao Wang

zwanggc@cse.ust.hk



Jiaxin Bai

jbai@connect.ust.hk



Yangqiu Song

yqsong@cse.ust.hk



Hanghang Tong

htong@illinois.edu



Knowledge Graph

❑ Represents a collection of interlinked descriptions of entities

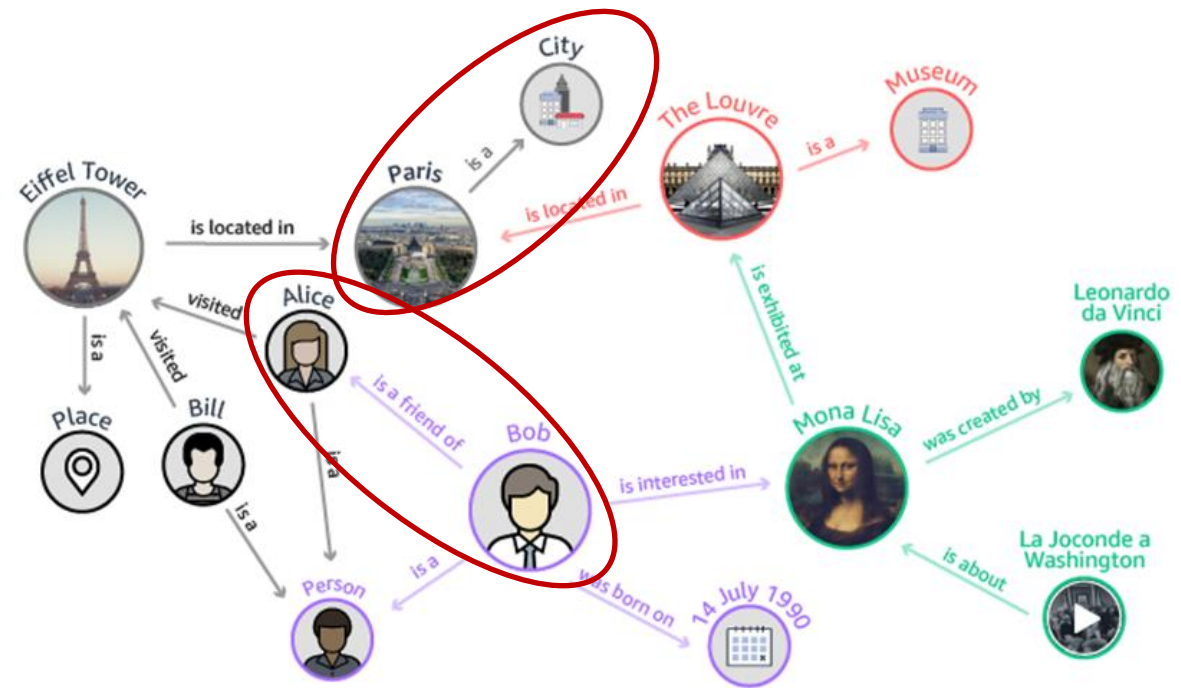
- Objects, events or concepts
- Multiple types of entities and relations exist

❑ Facts are represented as triplets (h, r, t)

- ('Paris', 'is_a', 'city')
- ('Alice', 'is_friend_of', 'Bob')
- ...

Notation & Symbols

- h : the head entity
- r : the relation
- t : the tail entity
- (h, r, t) : the triplet

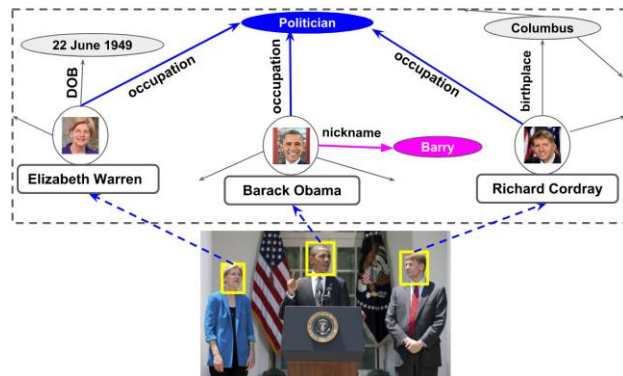


Abundance of Knowledge Graphs

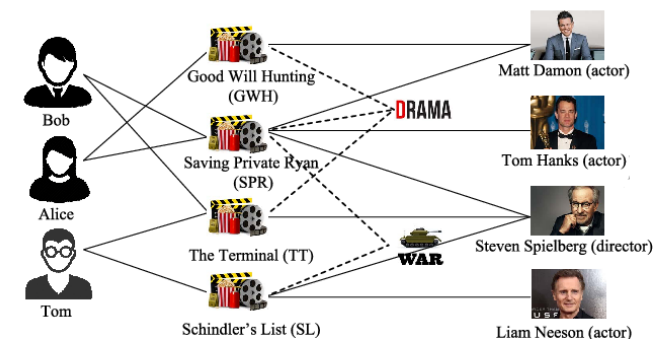
Knowledge Graph	Statements	Entities
 yago select knowledge	120 M	10 M
 WIKIDATA	610 M	51 M
 DBpedia	1.3 B	6 M
 GDELT	3.5 B	364 M

Knowledge Graph Applications

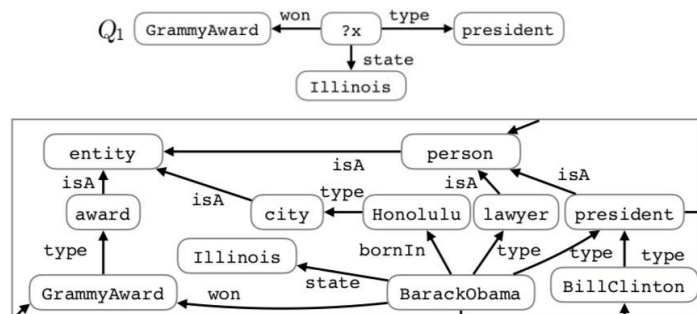
Computer Vision [1]



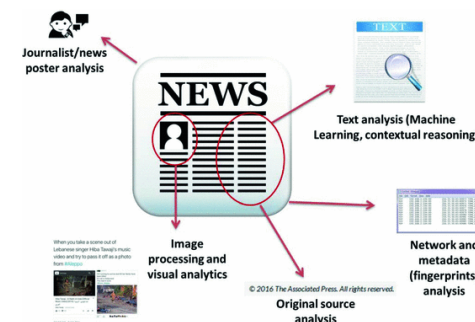
Recommendation [2]



Question Answering [3]



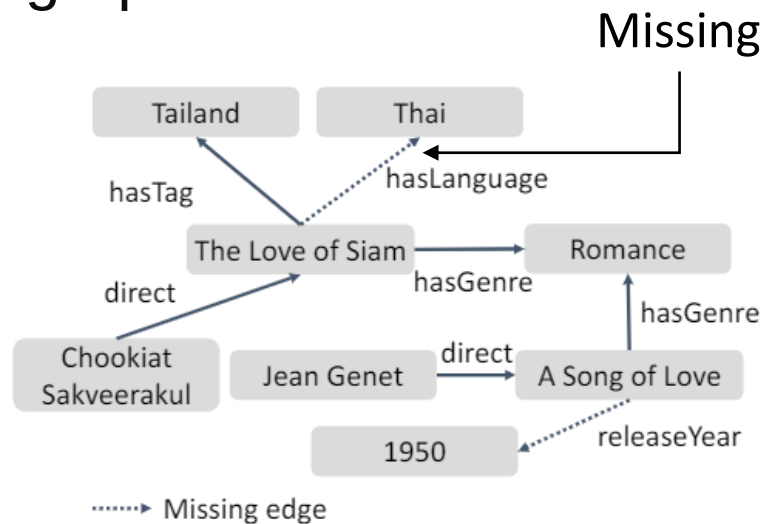
Fact Checking [4]



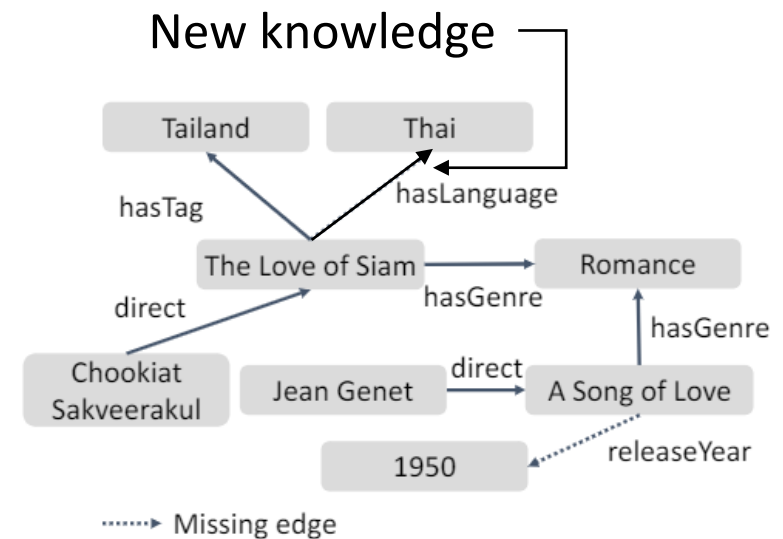
- [1] Y. Fang, K. Kuan, J. Lin, C. Tan, and V. Chandrasekhar. 2017. Object Detection Meets Knowledge Graphs(IJCAI 17).
- [2] F. Zhang, J. Yuan, D. Lian, X. Xie, and W. Ma. 2016. Collaborative Knowledge Base Embedding for Recommender Systems(KDD 16).
- [3] S. Hu, L. Zou, J. X. Yu, H. Wang, and D. Zhao. 2018. Answering Natural Language Questions by Subgraph Matching over Knowledge Graphs (TKDE 18).
- [4] P. Shiralkar, A. F. Flammini, F. Menczer, and G. Luca. 2017. Finding Streams in Knowledge Graphs to Support Fact Checking (ICDM 17).

What is Knowledge Graph Reasoning?

- ❑ **Goal:** **Infer or discover** new knowledge according to **existing information** in knowledge graphs in response to a query [1]
- ❑ **Benefit:** Enable models to learn from and reason with structured knowledge graph data



Infer
➔



- [1] X. Chen, S. Jia, and Y. Xiang. 2020. A review: Knowledge reasoning over knowledge graph (Expert Systems with Applications 20).

Knowledge Graph Reasoning Categorization

☐ Class: Deductive reasoning

- ☐ Apply **known rules** to derive knowledge

☐ (A, hasSon, B)&(B, hasSon, C)

☐ Infer: (A, isGrandFather, C)

☐ Class: Abductive reasoning

- ☐ Choose the best explanation that **explains an observation**

☐ (A, liveWith, B)

☐ Explanation: (A, hasSpouse, B)

☐ Class: Inductive reasoning

- ☐ Use patterns in observations to **derive knowledge**

☐ Give many examples of

- (A, hasSon, B)&(B, hasSon, C)
- (A, isGrandFather, C)

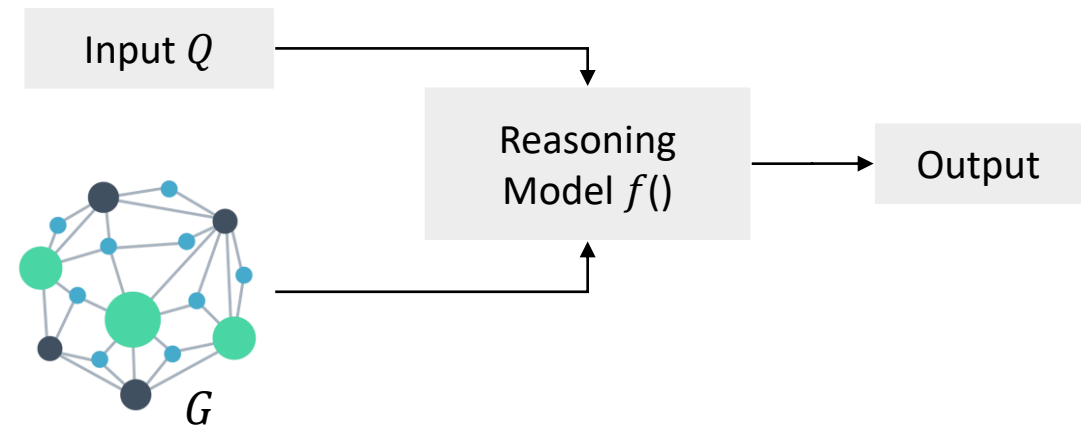
☐ Derive:

- hasSon & hasSon means isGrandFather

Knowledge Graph Reasoning Formulation

□ KG reasoning can be formulated as $f(Q, G)$

- f : the reasoning **function/model**
- Q : the reasoning **input/goal**
 - A partial triplet (h, r, ?)
 - Natural language question
 - Query graph
- G : the background **knowledge graph**



Overall Challenge #1: Background Knowledge Graph

❑ Size: KGs are **large**

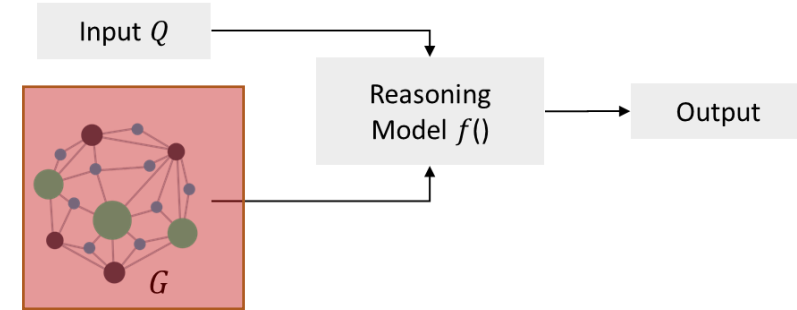
- For example, DBpedia contains 4.6 million entities
- Reasoning on large knowledge graph is time consuming

❑ Quality: KGs are **noisy and incomplete**

- For example, half of entities in DBpedia contain less than 5 relationships [1]
- Reasoning on incomplete knowledge graphs can be difficult

❑ Dynamics: Almost every knowledge graph **evolves over time** [2]

- Present an additional challenge



Overall Challenge #2: The Reasoning Model

❑ **Comprehensiveness:** Relations in KG have **different properties** [1]

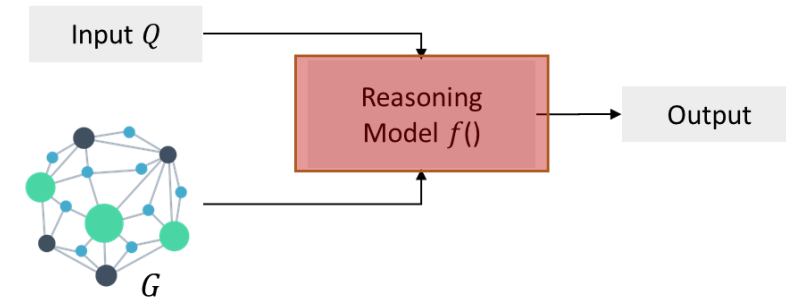
- Symmetry, antisymmetry, composition
- How to support these properties

❑ **Efficiency:** Knowledge graphs are large [2]

- How to reason **efficiently**

❑ **Generalizability:** Generalization ability of the model [3]

- Handle **new, unseen** data effectively



Overall Challenge #3: Reasoning Input

❑ Ambiguity: Q is ambiguous [1]

- Users are not familiar with the background KG, likely to ask ambiguous questions
 - “John Littel role in declaration independence” → role in : “**profession**” or “**occupation**”
 - “Thomas Jefferson role in declaration independence” → role in : “**film actor**”

❑ Interaction: Q is changing iteratively [2]

- Users can gradually refine the queries
 - “John Littel role in declaration independence” → “**Actor** John Littel role in **movie** declaration independence”
- User’s query intention may change correspondingly
 - 1. “**Who** directed Interstellar?”
 - 2. “**What other movies** did he also direct?”

- [1] L. Lihui, C. Yuzhong, D. Mahashweta, Y. Hao and T. Hanghang. 2023. Knowledge Graph Question Answering with Ambiguous Query (WWW 2023).
- [2] M. Kaiser, R. Saha Roy, and G. Weikum. 2021. Reinforcement Learning from Reformulations in Conversational Question Answering over Knowledge Graphs (SIGIR 21).

Roadmap

Part I: Knowledge Graph Reasoning: Basic Concepts and Techniques

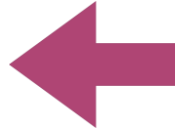
Part II: Recent Advance #1: Neural Reasoning for Natural Language Queries

Part III: Recent Advance #2: Neural Reasoning for Logical Queries

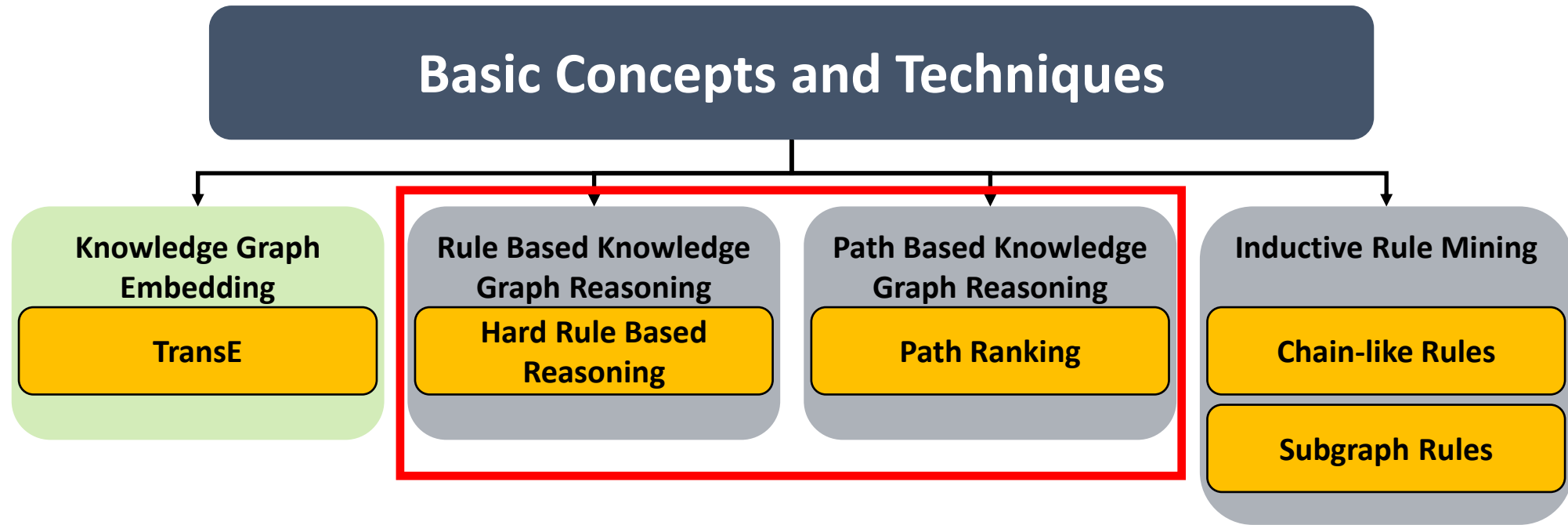
Part IV: Recent Advance #3: Neural Reasoning Beyond Entities and Relations

Part V: Recent Advance #4: LLM+KGR

Part VI: Open Challenges and Future Directions

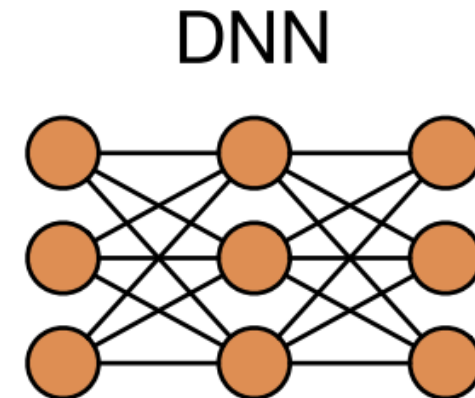
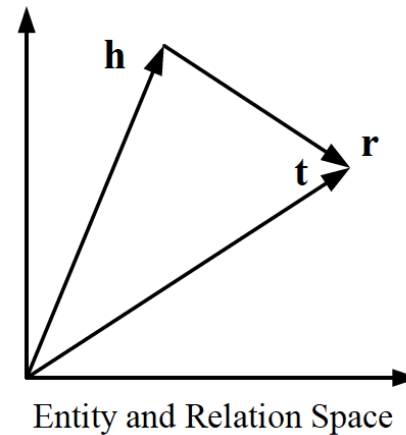
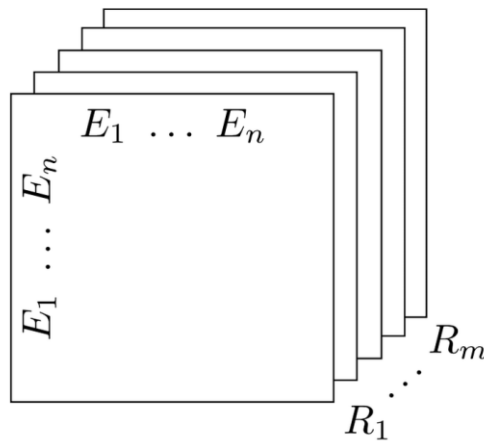


Overview of Part I



Reasoning in Continuous Space

- ❑ Reasoning based on low-dimensional vector representations
 - Knowledge graph embedding methods
- ❑ Project each entity and relation into a continuous vector space
- ❑ Tensor decomposition model [1], geometric model [2], deep learning model [3]



- [1] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G Bouchard. 2016. Complex Embeddings for Simple Link Prediction (ICML 16).
- [2] A. Bordes, N. Usunier, and A. Garcia-Duran. 2013. Translating Embeddings for Modeling Multi-relational Data (NeurIPS 13).
- [3] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings (AAAI 18).

Knowledge Graph Embedding

- ❑ **Goal:** Encode (1) entities as low-dimensional vectors and (2) relations as parametric algebraic operations in the continuous space
- ❑ **How-to:** Design a score function $f_r(\mathbf{h}, \mathbf{t})$ w.r.t. such embedding vectors so that a true triplet receives higher score than a false one
- ❑ **KGE design rationale:** Capture KG patterns
 - Symmetry, antisymmetry, inversion and composition
- ❑ **Applications of knowledge graph embedding**
 - Knowledge graph completion
 - Question answering
 - Recommender system

Notation & Symbols

- h : head entity
- r : relation
- t : tail entity
- $f_r(\mathbf{h}, \mathbf{t})$: the score function
- $d_r(\mathbf{h}, \mathbf{t})$: the distance function
- True/positive triplet: (h, r, t)
- False/negative triplet: $(h', r, t), (h, r, t'), (h', r, t')$
- $\mathbf{h}$: head entity embedding
- $\mathbf{r}$: relation embedding
- $\mathbf{t}$: tail entity embedding

- [1] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G Bouchard. 2016. Complex Embeddings for Simple Link Prediction (ICML 16).
- [2] A. Bordes, N. Usunier, and A. Garcia-Duran. 2013. Translating Embeddings for Modeling Multi-relational Data (NeurIPS 13).
- [3] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings (AAAI 18).

Preliminary: Symmetric/Antisymmetric Relations

□ Symmetric/Antisymmetric Relations

- **Symmetric:** e.g., Marriage
- **Antisymmetric:** e.g., hasChild

□ Formally:

r is **Symmetric**: $r(x, y) \Rightarrow r(y, x)$ if $\forall x, y$

r is **Antisymmetric**: $r(x, y) \Rightarrow \neg r(y, x)$ if $\forall x, y$

Preliminary: Inverse Relations

□ Inverse Relations

- Hypernym and hyponym:
 - Color is the hypernym (r_2) of blue, and blue is the hyponym (r_1) of color
- Husband (r_2) and wife (r_1)

□ Formally:

r_1 is inverse to relation r_2 : $r_2(x, y) \Rightarrow r_1(y, x)$ if $\forall x, y$

Preliminary: Composition Relations

□ Composition Relations

- My mother's husband is my father
- r_1 : hasMother, r_2 : hasHusband
- r_3 : hasFather

□ Formally:

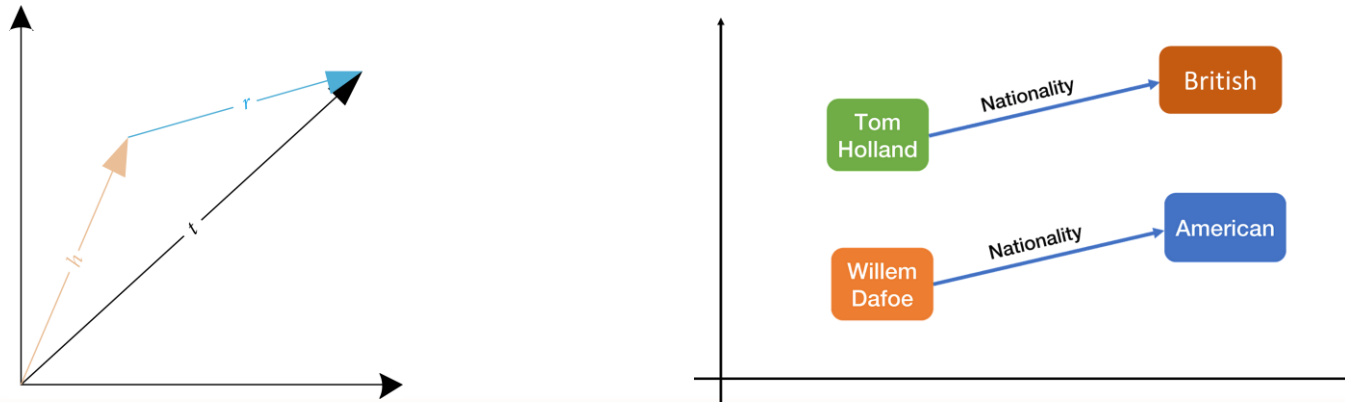
r_3 is a **composition** of relation r_1 and relation r_2 :

$$r_1(x, y) \wedge r_2(y, z) \Rightarrow r_3(x, z) \text{ if } \forall x, y, z$$

KG Embedding Method #1: TransE

- ❑ **Embedding space:** Each entity and relation as a low-dimensional vector in R^k
- ❑ **Key idea:** Relation r as a translation from the head entity h to the tail entity t
 - An ideal/predicted tail entity: $\mathbf{t}_{\text{pred}} = \mathbf{h} + \mathbf{r}$
- ❑ **Score function:** $f_r(\mathbf{h}, \mathbf{t}) = -d_r(\mathbf{h}, \mathbf{t}) = -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$
- ❑ **Distance function:** $d_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$

- Triplet: $(\mathbf{h}, \mathbf{r}, \mathbf{t})$
- Embedding vectors: $\mathbf{h}, \mathbf{r}, \mathbf{t}$



• [1] A. Bordes, N. Usunier, and A. Garcia-Duran. 2013. Translating Embeddings for Modeling Multi-relational Data (NeurIPS 13).

TransE: Training Process

□ For each positive triplet $(h, r, t) \in S$,

○ Sample a set of corrupted triplets $(h, r, t') \in S'_{(h,r,t)}$ or $(h', r, t) \in S'_{(h,r,t)}$

□ Learning: a margin-based ranking loss

- True triplet: (h, r, t)
- Corrupted triplets:
 - $(h', r, t), (h, r, t'), (h', r, t')$
- γ : the margin

□ Minimize $L = \sum_{(h,r,t) \in S} \sum_{(h',r,t') \in S'_{(h,r,t)}} \max(\gamma + d_r(\mathbf{h}, \mathbf{t}) - d_r(\mathbf{h}', \mathbf{t}'), 0)$

Margin: gap should be at least γ

TransE: Key Properties

❑ Pros

- Can model antisymmetric relations: $\mathbf{h} + \mathbf{r} = \mathbf{t}$, but $\mathbf{t} + \mathbf{r} \neq \mathbf{h}$ if $\mathbf{r} \neq 0$
- Can model inverse relations: $\mathbf{h} + \mathbf{r}_1 = \mathbf{t}$, $\mathbf{t} + \mathbf{r}_2 = \mathbf{h}$, $\mathbf{r}_1 = -\mathbf{r}_2$
- Can model composition relations: $\mathbf{r}_3 = \mathbf{r}_1 + \mathbf{r}_2$

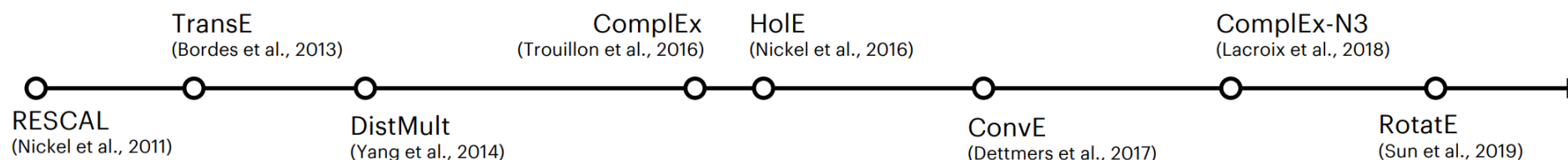
❑ Cons

- Cannot model symmetric relations: $\mathbf{h} + \mathbf{r} = \mathbf{t}$, $\mathbf{t} + \mathbf{r} = \mathbf{h}$, then $\mathbf{r} = 0$

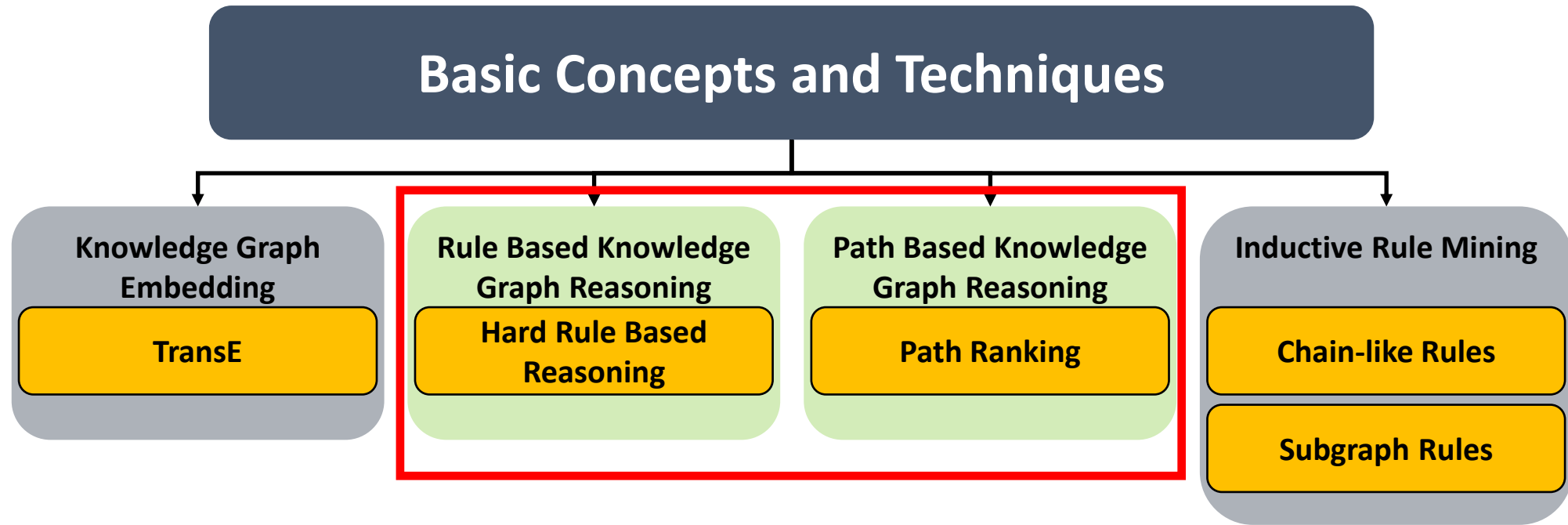
Knowledge Graph Embedding: Summary

Model	Score Function	Symmetry	Antisymmetry	Inversion	Composition
TransE [2]	$- \mathbf{h} + \mathbf{r} - \mathbf{t} $	✗	✓	✓	✓
DistMult [3]	$\langle \mathbf{h}, \mathbf{r}, \mathbf{t} \rangle$	✓	✗	✗	✗
ComplEx [4]	$\text{Re}(\langle \mathbf{h}, \mathbf{r}, \bar{\mathbf{t}} \rangle)$	✓	✓	✓	✗
RotatE [8]	$- \mathbf{h} \circ \mathbf{r} - \mathbf{t} $	✓	✓	✓	✓

(Some) KGE models in recent literature:



Overview of Part I



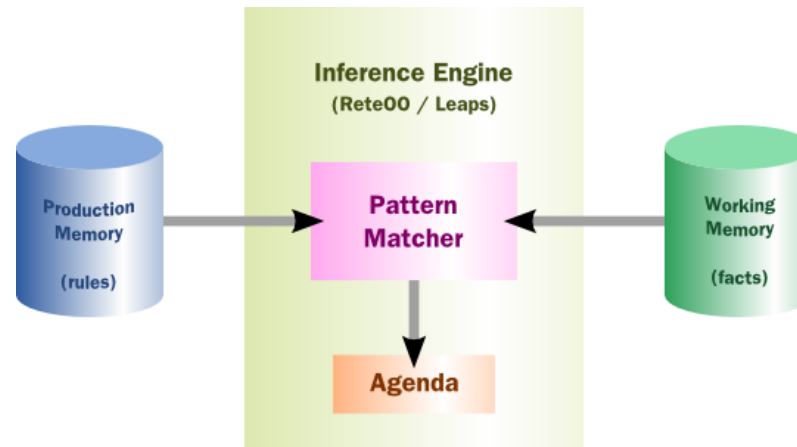
Rule-based Expert System

❑ **Key idea:** Apply rules **iteratively** to generate new facts

- New facts represent conclusions about the state of the domain given the observations

❑ **Major components**

- The **inference engine**
- The **knowledge bases** and **rules**



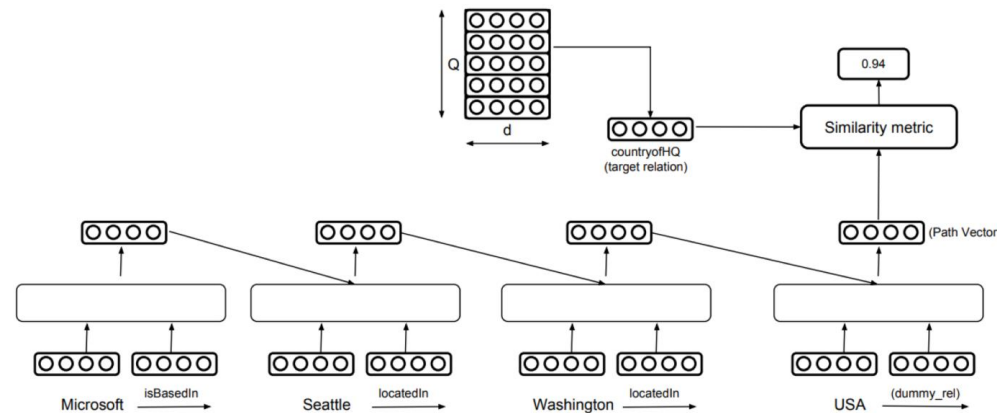
- [1] H. Roth, F. Waterman, and D. Arthur. 1936. Building expert systems (Addison-Wesley 36).

Inference Engine

- ❑ Brain of the reasoning system
- ❑ Inferring knowledge by applying forward or backward chaining
 - Forward chaining (data driven)
 - Start with facts, determine applicable rules, and apply one
 - Repeat until no more rules can be applied
 - Backward chaining (goal oriented) :
 - Give a goal, rules are applied by matching the goal to infer the answer
 - E.g. (Alan Turing, wasBorn, ?)

Path Based Knowledge Graph Reasoning

- ❑ Logical rule reasoning
 - Require given logic rules as input
- ❑ Path-based reasoning
 - Involves traversing paths in a KG to infer new information
 - Paths can be directed or undirected



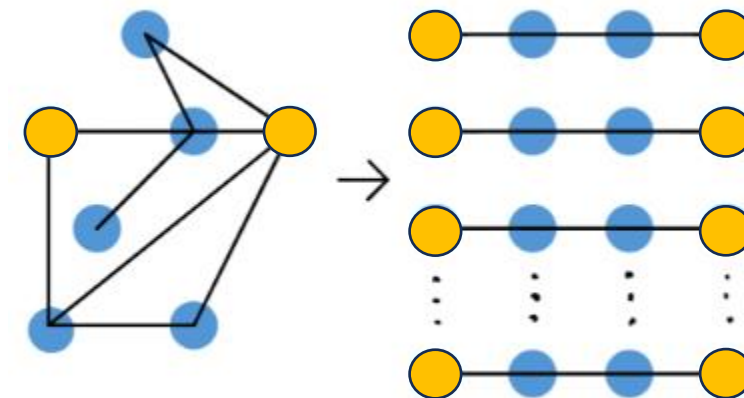
- [1] N. Lao, T. Mitchell, and W. Cohen. 2011. Random walk inference and learning in a large scale knowledge base (ACL 11).
- [2] H. Wang, H. Ren, and J. Leskovec. 2021. Relational message passing for knowledge graph completion (KDD 21).
- [3] W. Xiong, T. Hoang, and WY. Wang. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning (ACL 17).

PRA: Reasoning with Random Walks

❑ **Assumption:** random walks $\rightarrow$ relational features

❑ **Key ideas**

- A relation path $p = (R_1, \dots, R_n)$ is a sequence of relations
- Run random walk to derive many paths
- Use supervised training to predict the score of triplet



	Path 1	Path 2	...	Path n	Label
Query 1	Score 1.1	Score 1.2	...	Score 1.n	y1
Query 2	...	...	...	...	y2
...	...	...	...	...	...
Query k	Score k.1	Score k.2	...	Score k.n	yk

- [1] N. Lao, T. Mitchell, and W. Cohen. 2011. Random walk inference and learning in a large scale knowledge base (ACL 11).

Related works

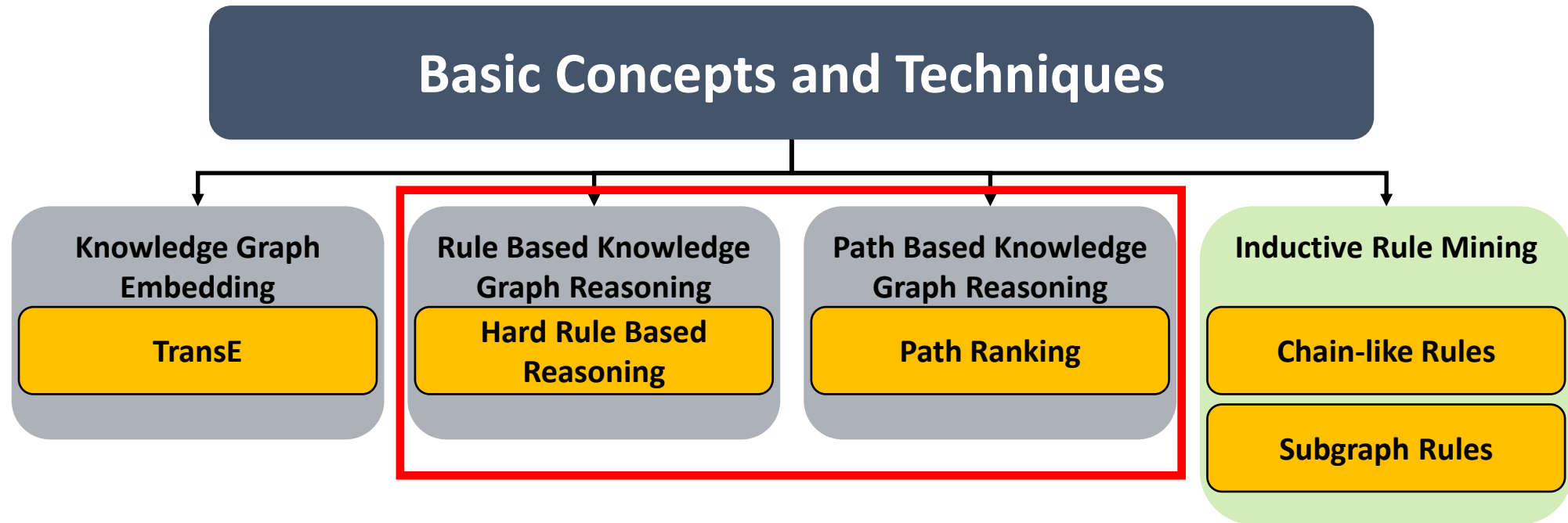
□ Rule based knowledge graph reasoning

- M. Richardson, and P. Domingos. 2006. Markov Logic Networks (Machine learning 16).
- W. Cohen. 2016. TensorLog: A Differentiable Deductive Database (arXiv 16).
- P. Cheng, P. Liu, P. Wang, P. Sun. 2022. RLogic: Recursive Logical Rule Learning from Knowledge Graphs (KDD 22).
- ...

□ Path based knowledge graph reasoning

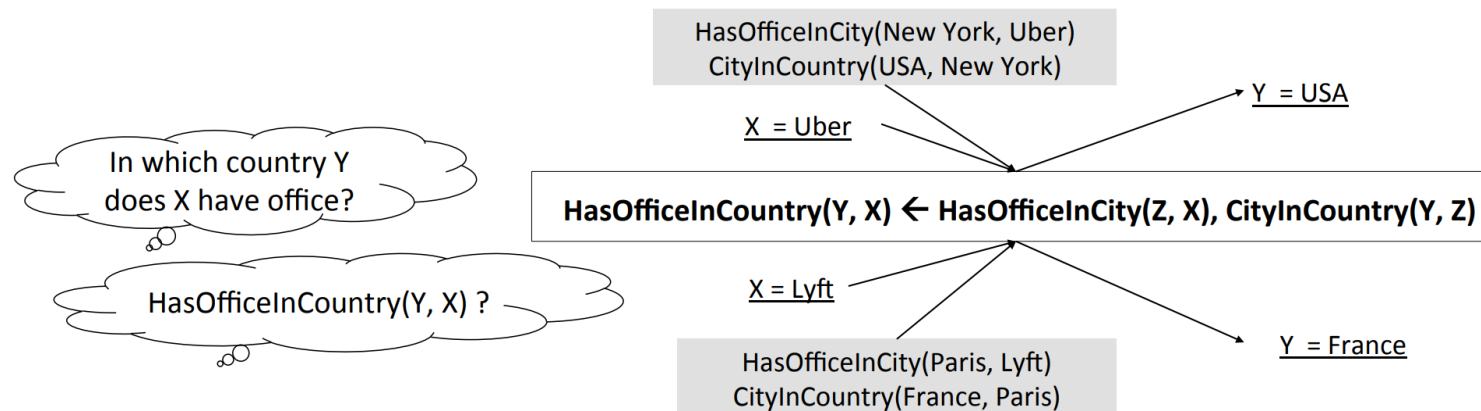
- H. Wang, H. Ren, and J. Leskovec. 2021. Relational message passing for knowledge graph completion (KDD 21).
- W. Xiong, T. Hoang, and WY. Wang. 2017. DeepPath: A reinforcement learning method for knowledge graph reasoning (ACL 17).
- M. Qu, J. Chen, J. Tang. 2021. RNNLogic: Learning logic rules for reasoning on knowledge graphs (ICLR 21).
- ...

Overview of Part I



Why Mining Logical Rules?

- ❑ **Goal:** Extract general and meaningful rules that can be applied to new, unseen data.
- ❑ **Advantages of mining logical rules**
 - Logical rules have the potential to generalize well
 - Logical rules are explainable and understandable

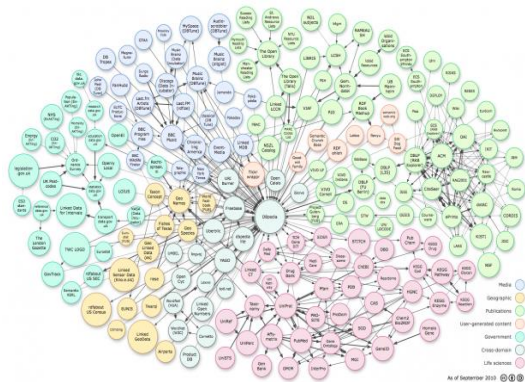


- [1] F. Yang, Z. Yang, W. Cohen. 2017. Differentiable Learning of Logical Rules for Knowledge Base Reasoning (NeurIPS 17).

Inductive Rule Mining

□ Process of inductive rule mining

- Rule generation: Use a rule mining algorithm to discover interesting rules from the dataset
- Rule evaluation: Assess the quality and relevance of the generated rules.



$\text{Appears_in_TV_Show}(X, Y) \leftarrow \text{Has_Actor}(X, Y)$
 $\text{Appears_in_TV_Show}(X, Y) \leftarrow \text{Creator_of}(X, U) \wedge \text{Has_Producer}(U, V) \wedge \text{Appears_in_TV_Show}(V, Y)$

$\text{ORG_in_State}(X, Y) \leftarrow \text{ORG_in_City}(X, U) \wedge \text{City_Locates_in_State}(U, Y)$
 $\text{ORG_in_State}(X, Y) \leftarrow \text{ORG_in_City}(X, U) \wedge \text{Address_of_PERS.}(U, V) \wedge \text{Born_in}(V, W) \wedge \text{Town_in_State}(W, Y)$

$\text{Person_Nationality}(X, Y) \leftarrow \text{Born_in}(X, U) \wedge \text{Place_in_Country}(U, Y)$
 $\text{Person_Nationality}(X, Y) \leftarrow \text{Student_of_Educational_Institution}(X, U) \wedge \text{ORG_Endowment_Currency}(U, V) \wedge \text{Currency_Used_in_Region}(V, W) \wedge \text{Region_in_Country}(W, Y)$

- [1] M. Qu, Z. Zhu, J. Tang. 2022. Reasoning on Knowledge Graphs: Symbolic or Neural? (AAAI 22).

Neural LP

□ Goal: learn logical rules with confidence scores (alpha)

$$\alpha \text{ query } (Y, X) \leftarrow R_n (Y, Z_n) \wedge \cdots \wedge R_1 (Z_1, X)$$

□ TensorLog

- Reasoning by matrix multiplication
- Denote each entity as a one-hot vector
- Denote each relation as a matrix R : the (i, j) -entry is 1 if only if $R(i, j)$ is a fact in KG
- During reasoning, for a rule $R_1(X, Z) \wedge R_2(Z, Y) \rightarrow R_3(X, Y)$ and query $R_3(X, ?)$
 - Computer $R_1(X, Z) \wedge R_2(Z, Y) = a M_{R_2} M_{R_1} v_X$
 - Retrieving entities whose entries are non-zeros as answers

Neural LP: Key Ideas

□ From length 2 to length l :

$$\circ R_1(X, Z) \wedge R_2(Z, Y) = M_{R_2} M_{R_1} v_X \rightarrow R_1(X, Z) \wedge \dots \wedge R_l(Q, Y) = \sum_l \alpha_l \prod_{k \in \beta_l} M_{R_k} v_X$$

□ Objective function of all entity pairs

$$s = \sum_l (\alpha_l (\prod_{k \in \beta_l} M_{R_k} v_x)) , \text{score}(y | x) = v_y^T s$$

$$\max_{\{\alpha_l, \beta_l\}} \sum_{\{x, y\}} \text{score}(y | x) = \max_{\{\alpha_l, \beta_l\}} \sum_{\{x, y\}} v_y^T \left(\sum_l (\alpha_l (\prod_{k \in \beta_l} M_{R_k} v_x)) \right)$$

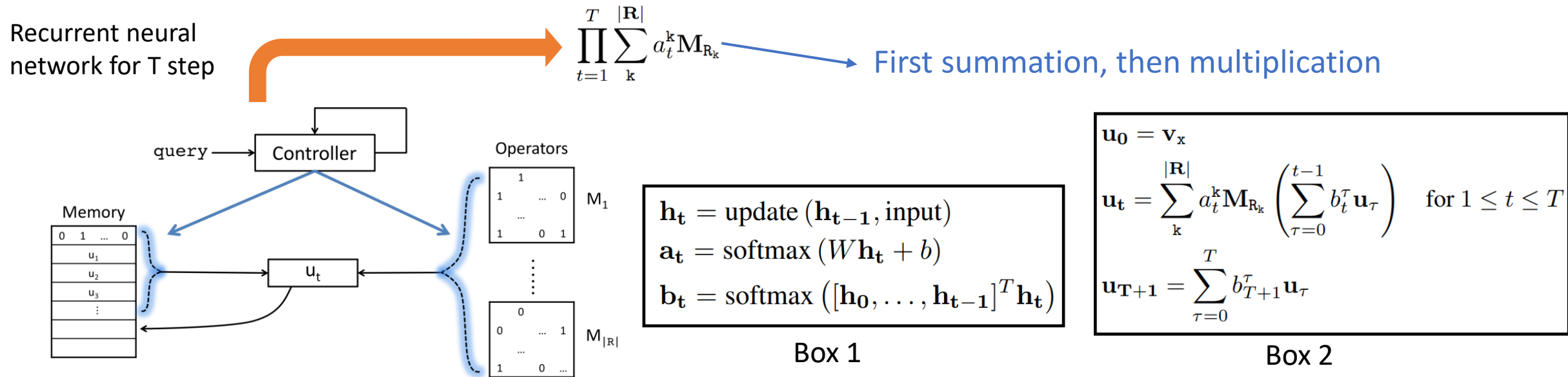
□ Differentiable rules

- Exchange product and sum: $\sum_l \alpha_l \prod_{k \in \beta_l} M_{R_k} \rightarrow \prod_{t=1}^T \sum_k a_t^k M_{R_k}$
- Now learning a single rule, **each step is combination of relationships**

Neural LP: Key Ideas

□ Differentiable rules

- Now learning a single rule, each step is combination of relationships



- [1] F. Yang, Z. Yang, W. Cohen. 2017. Differentiable Learning of Logical Rules for Knowledge Base Reasoning (NeurIPS 17).

Grail: Subgraph Reasoning

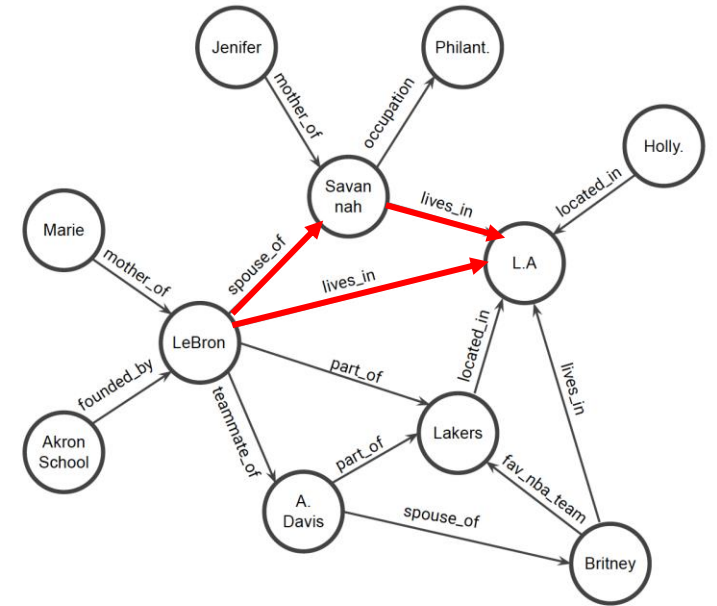
□ Observation

- Useful rules contained in the subgraph around query

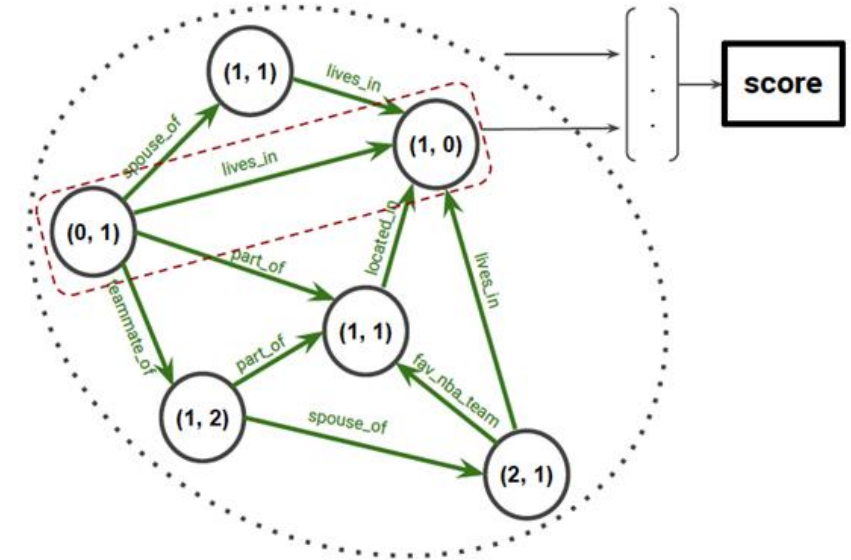
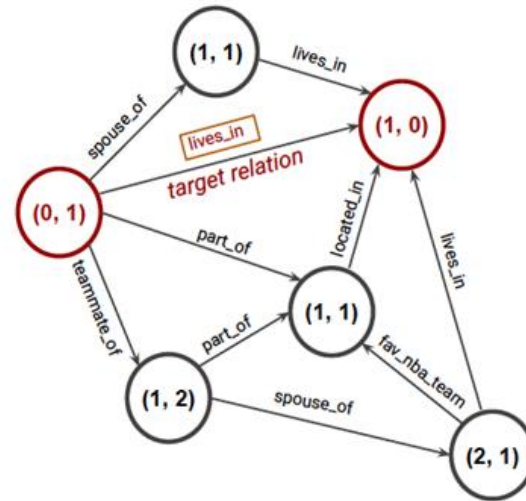
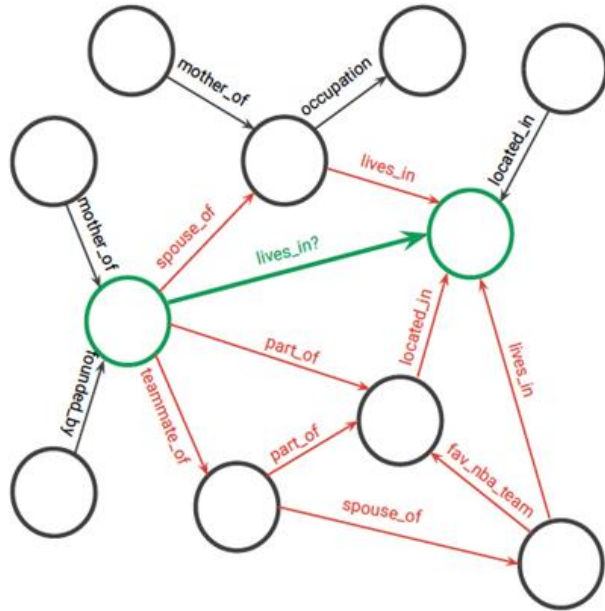
$$(X, \text{lives_in}, Y) \leftarrow \exists Z. (X, \text{spouse_of}, Z) \wedge (Z, \text{lives_in}, Y)$$

□ Key ideas:

- Idea 1: Apply graph neural networks (GNNs) on the subgraphs surrounding candidate edge.
- Idea 2: Subgraph reasoning avoids explicit rule induction.
- Idea 3: Ensure model is expressive enough to capture logical rules.



Grall: Example



1. Extract subgraph around candidate edge

2. Assign structural labels to nodes

3. Run GNN on the extracted subgraph

- [1] K. Teru, E. Denis, W.L. Hamilton. 2020. Inductive Relation Prediction by Subgraph Reasoning (ICML 20).

Roadmap

Part I: Knowledge Graph Reasoning: Basic Concepts and Techniques

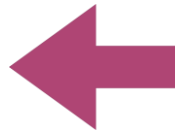
Part II: Recent Advance #1: Neural Reasoning for Natural Language Queries

Part III: Recent Advance #2: Neural Reasoning for Logical Queries

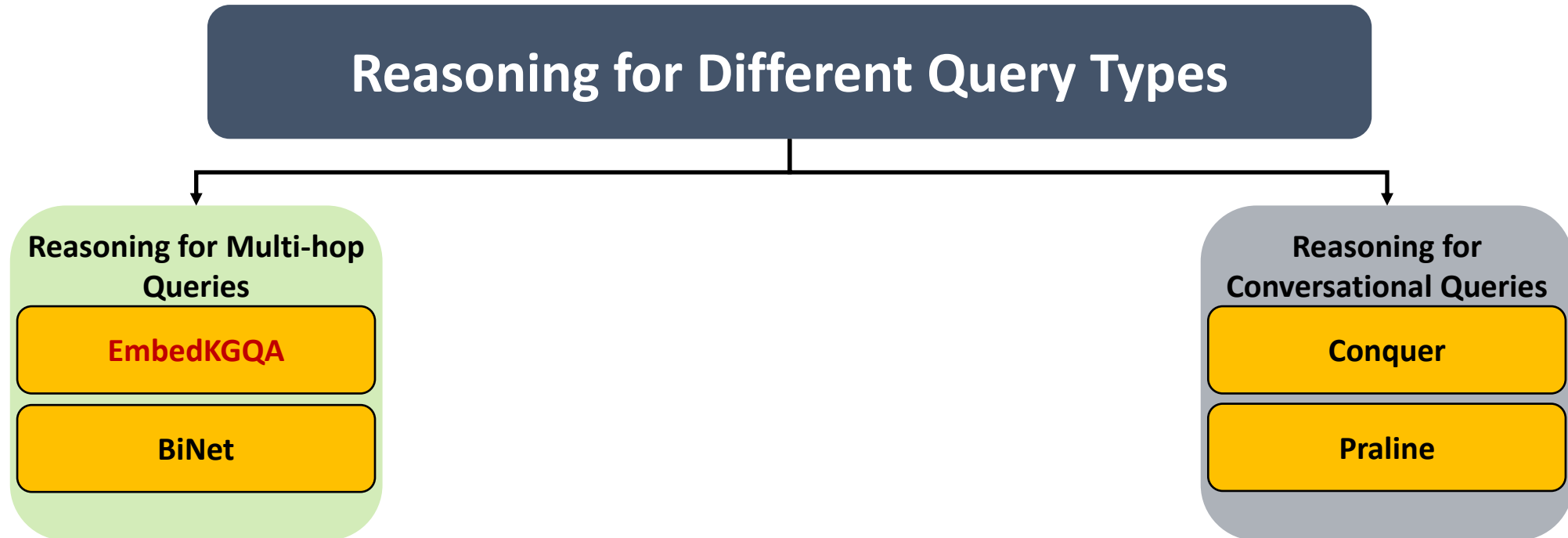
Part IV: Recent Advance #3: Neural Reasoning Beyond Entities and Relations

Part V: Recent Advance #4: LLM+KGR

Part VI: Open Challenges and Future Directions



Overview of Part II



Preliminary: KGQA

□ KGQA definition

- Given: (1) a natural language question (2) the topic entity, and (3) a knowledge graph
- Find: a set of nodes of the knowledge graph to answer the question

□ Multi-hop question (the focus of this part)

- Can be transformed to a path on KG
- Example: “what is the language of the film directed by Steven Spielberg?”

□ Other types of questions

- One-hop question: “What is the genre of Interstellar?”
- Logic query: “Where did Canadian citizens with Turing Award graduate?”
- Many more

- [1] H. Ren, W. Hu, J. Leskovec. 2020. Query2box: Reasoning over Knowledge Graphs in Vector Space using Box Embeddings (ICLR 20).
- [2] H. Xiao, J. Zhang, D. Li, P. Li. 2019. Knowledge Graph Embedding Based Question Answering (WSDM 19).
- [3] A. Saxena, A. Tripathi, P. Talukdar. 2020. Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings (ACL 20)

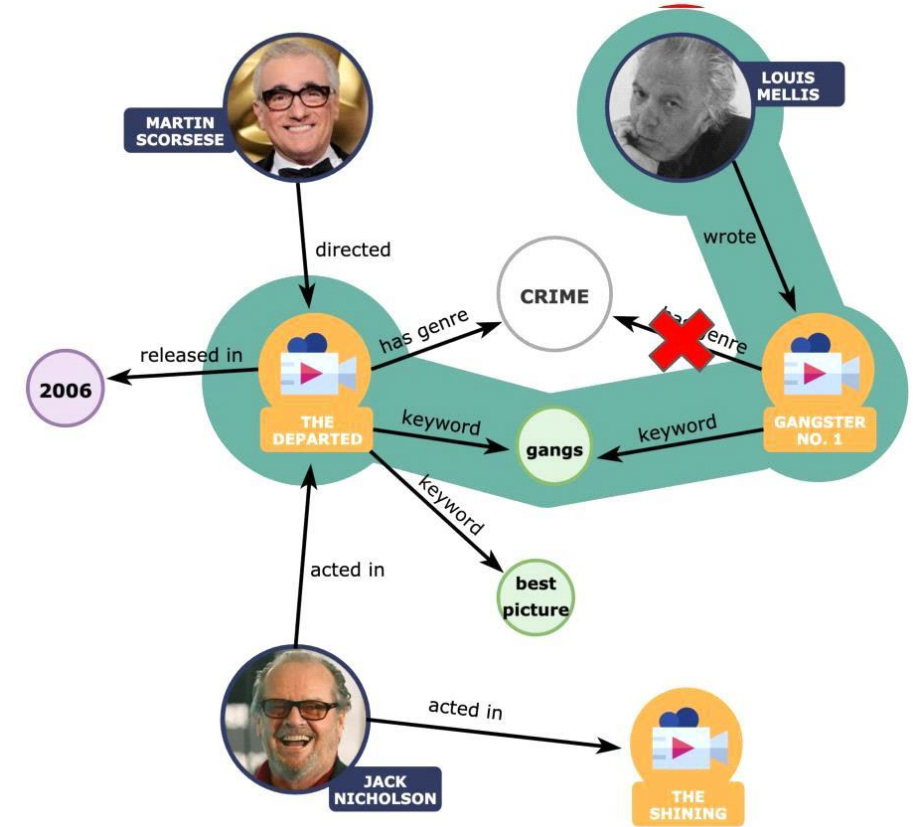
Key Challenge: KG Sparsity & Incompleteness

❑ Knowledge graphs are sparse & incomplete

- Break of the reasoning chain
- Increase the length of reasoning path

❑ Possible solutions

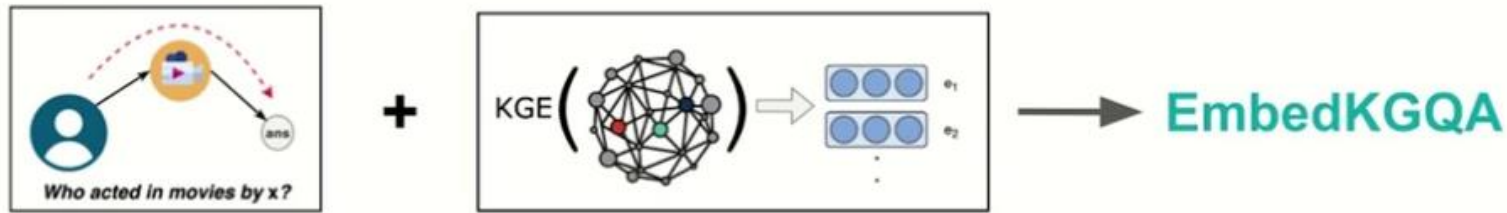
- #1: Use additional data e.g., text corpus
 - But text not always available
- #2: Impose neighborhood limits
 - But the answer might be out of reach.



Q: What's the genre of movies directed by Louis Mella?

EmbedKGQA: Overview

□ Key ideas

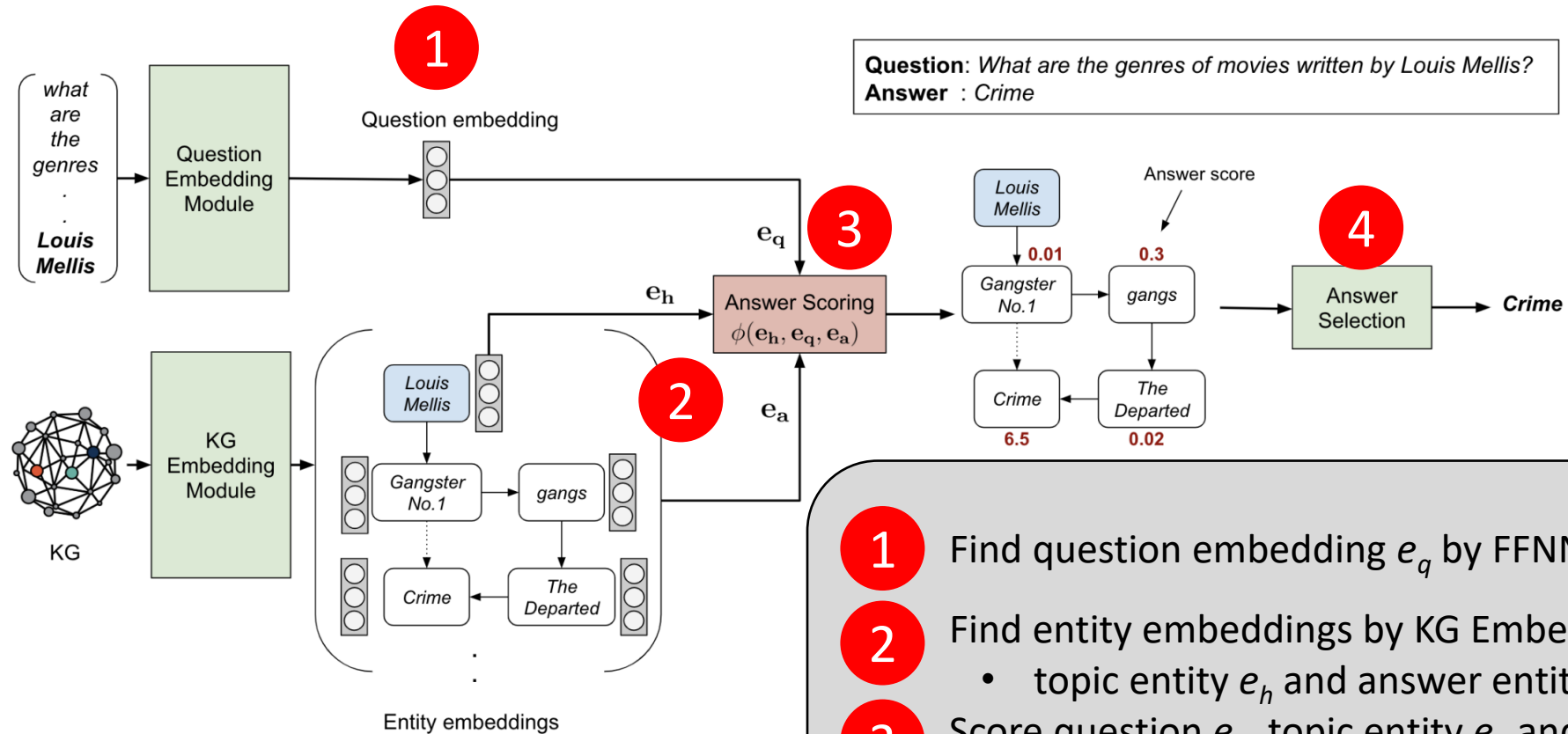


- Embeds question & the entities of the KG in a vector space
- Performs QA using these embeddings

□ Benefits

- Deal with KG sparsity & incompleteness
- Bypass the neighborhood limits

EmbedKGQA: Details



- 1 Find question embedding e_q by FFNN
- 2 Find entity embeddings by KG Embedding (ComplEx, See Part 1)
 - topic entity e_h and answer entity e_a
- 3 Score question e_q , topic entity e_h and answer e_a (ComplEx)
 - $\phi(e_q, e_h, e_a)$
- 4 Select answer $\hat{a} = \operatorname{argmax}_a \phi(e_q, e_h, e_a)$

• [1] A. Saxena, A. Tripathi, P. Talukdar. 2020. Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings (ACL 20)

KG Completion vs. EmbedKGQA

❑ KGC (predict missing tail entity)

○ Problem definition

- Given: the head h , the relation r , and KG
- Predict: missing tail entity

○ How-to (see Part 1)

- Choose a score function $f_r(h, t)$
- Find embeddings for all entities and relations
- Predict the tail entity $\hat{t} = \operatorname{argmax}_t f_r(h, t)$

❑ EmbedKGQA (answer multi-hop question)

○ Problem definition

- Given: the question, its topic entity, and KG
- Predict: missing answer entity

○ How-to

- Find question embedding e_q by FFNN
- Choose a score function $\phi(e_q, e_h, e_a)$
- Find embeddings for all entities
- Predict the answer entity $\hat{a} = \operatorname{argmax}_a \phi(e_q, e_h, e_a)$

EmbedKGQA is essentially a KGC task to predict the missing tail entity, by

- Treating (1) the topic entity as the head entity, and (2) the question as a special type of relation
- Finding the question/relation embedding by FFNN (not via KG embedding)

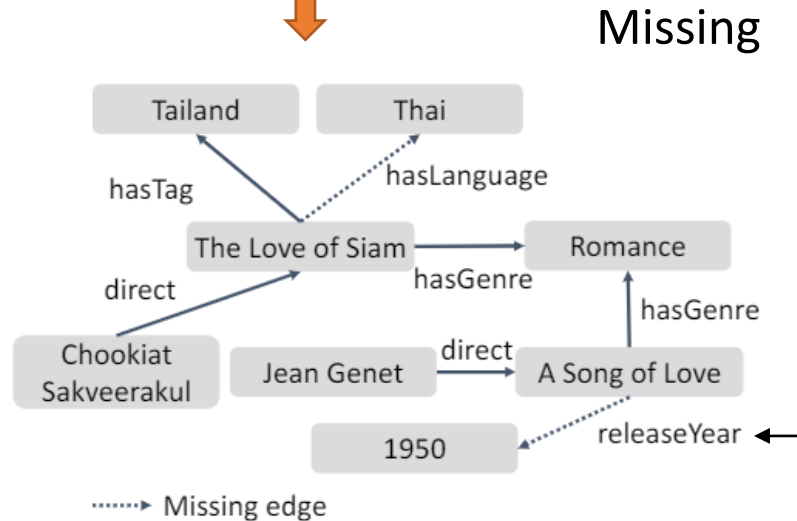
• [1] A. Saxena, A. Tripathi, P. Talukdar. 2020. Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings (ACL 20)

Insight #1: KGQA can Help KGC

❑ Want to predict (A Song of Love, releaseYear, ?)



Hard to answer



"which years were all the films directed by Jean Genet released?"
Answer: 1950

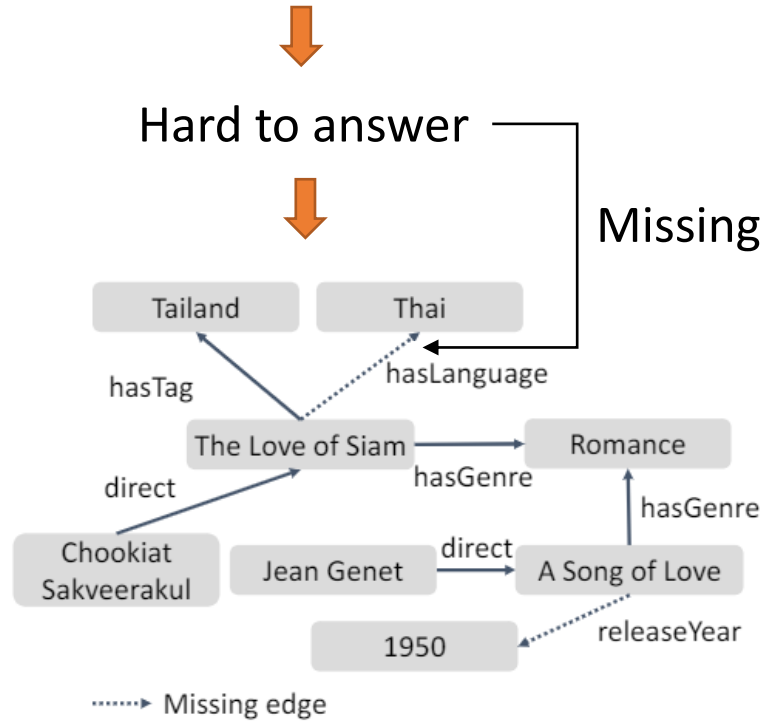


Predict: (A Song of Love, releaseYear, 1950)
Because Jean Genet **directed** "A Song of Love"
And Jean Genet **only directed** one movie in his life

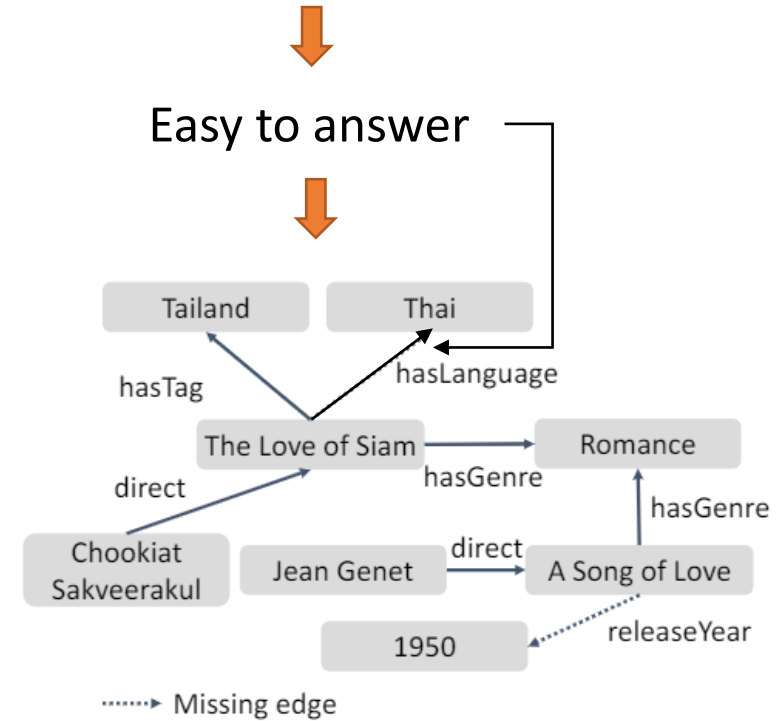
- [1] L. Liu, B. Du, J. Xu, Y. Xia, H. Tong. 2022. Joint Knowledge Graph Completion and Question Answering (KDD 22).

Insight #2: KGC can Help KGQA

❑ Question: “what is the language of the film The Love of Siam”



After completion



- [1] L. Liu, B. Du, J. Xu, Y. Xia, H. Tong. 2022. Joint Knowledge Graph Completion and Question Answering (KDD 22).

Theoretical Analysis

Main claim/insight: Knowledge graph completion is a special case of multi-hop question answering.

❑ Knowledge graph completion: $Pr(t | h, r_1)$

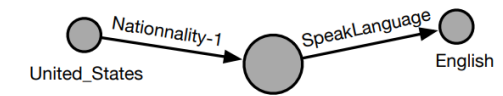
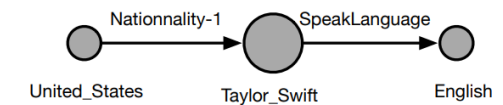
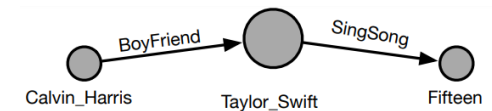
❑ Multi-hop question answering:

$$\circ Pr(t | h, r_1, r_2, \dots, r_n) = Pr(a_1 | h, r_1) Pr(a_2 | a_1, r_2) \dots Pr(t | a_{n-1}, r_n)$$

❑ Maximize $Pr(t | h, r_1) \rightarrow \text{maximize } Pr(t | h, r_1, r_2, \dots, r_n)$

❑ Maximize $Pr(t | h, r_1, r_2, \dots, r_n) \rightarrow Pr(t | h, r_1)$

❑ Question: whether can jointly perform KGC & KGQA in a mutually beneficial way?



- [1] Apoorv Saxena et.al. 2020. Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings (ACL 20)
- [2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data(NIPS 13).

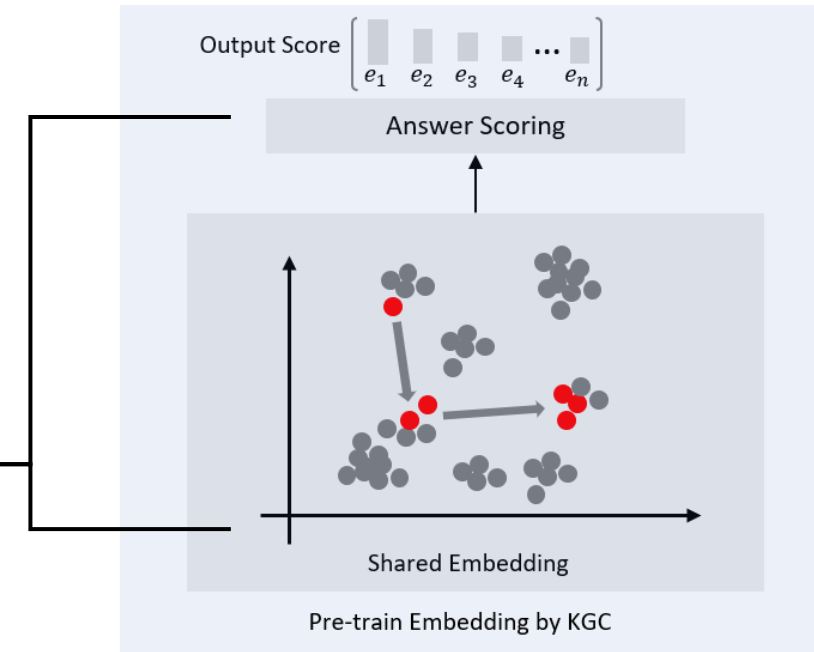
BiNet: Key Ideas

❑ Question #1: How can KGC help KGQA

❑ Our solution: **Share latent features**

- Formulate it as a multi-task learning problem
- Pre-train embedding by KGC
- Multi-hop KGQA and KGC
 - Share the embedding space and the answer scoring module
 - Automatically share latent features and reinforce each other

Shared by KGC and KGQA

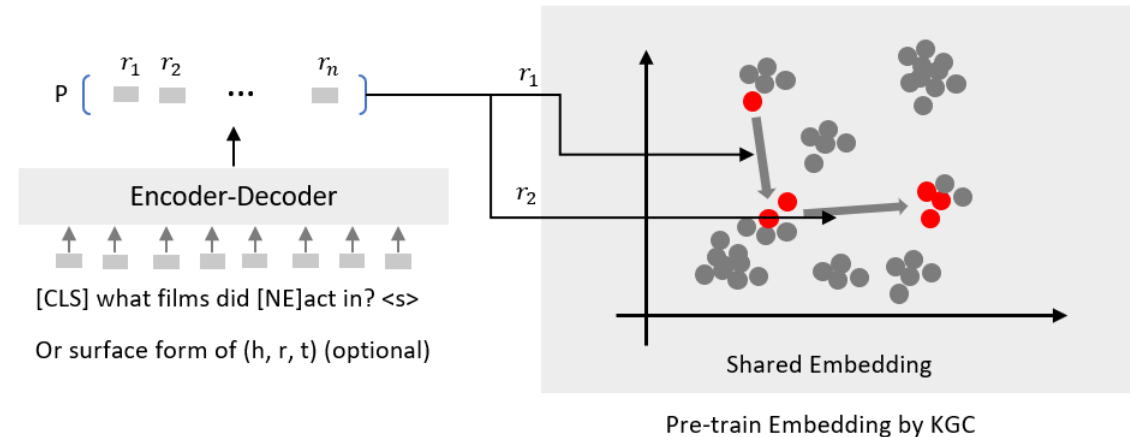


BiNet: Key Ideas

❑ Question #2: How can KGQA help KGC

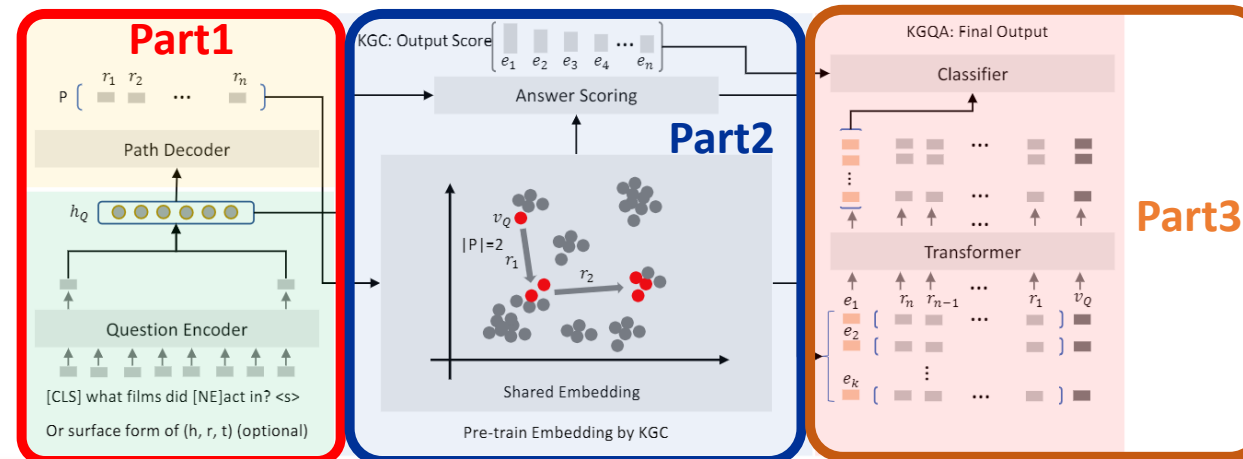
❑ Our solution: **Path based completion**

- **Transform** the natural language question **to a path** in the knowledge graph
- **Complete** the knowledge graph **according to (v, P, a)**
 - v : topic entity in the natural language question
 - P : the path
 - a : the answer



BiNet: Model Overview

- ❑ **Part1 (Path Decoder):** Using an Encoder-Decoder model to transform the natural language to a path P
- ❑ **Part2 (Answer Scoring):** Using P to complete the knowledge graph. Finding k answer candidates by the answer scoring module
- ❑ **Part3 (Answer Refinement):** Re-rank answers by Transformer Encoder



• [1] L. Liu, B. Du, J. Xu, Y. Xia, H. Tong. 2022. Joint Knowledge Graph Completion and Question Answering (KDD 22).

BiNet: Experiment

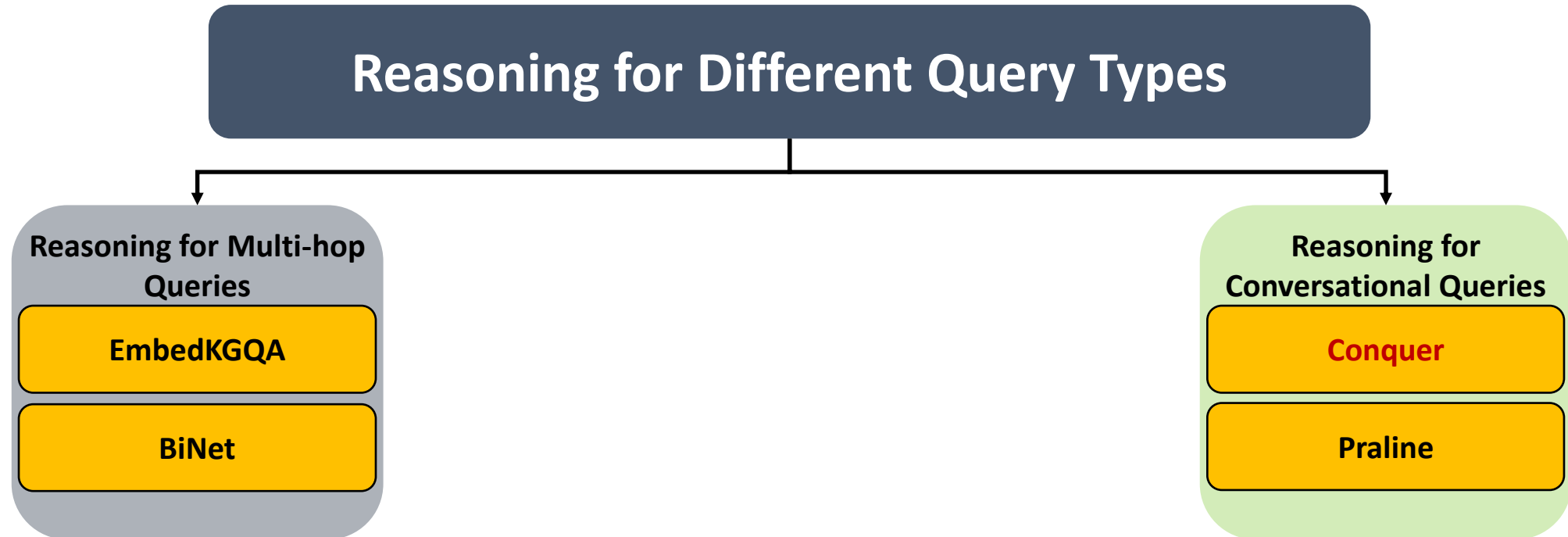
□ MetaQA: Question answering accuracy

	50% KG				30% KG			
Model	MetaQA-1	MetaQA-2	MetaQA-3	Avg	MetaQA-1	MetaQA-2	MetaQA-3	Avg
GraftNet	64.0	52.6	59.2	58.6	48.4			48.4
PullNet	65.1	52.1	59.7	59.0	-	-	-	-
KV-Mem	63.6	41.8	37.6	47.7	44.7			44.7
EmbedKGQA	83.1	91.8	70.3	81.7	77.7	81.2	69.0	76.0
BiNet	84.2	92.8	75.9	84.3	77.8	86.4	74.3	79.5

□ Observations

- An average advantage of 2.5% on 50% KG and an average advantage of 3.5% on 30% KG
- For sparse KG, BiNet outperforms baseline more
- For longer path questions, BiNet outperforms baselines more

Overview of Part II

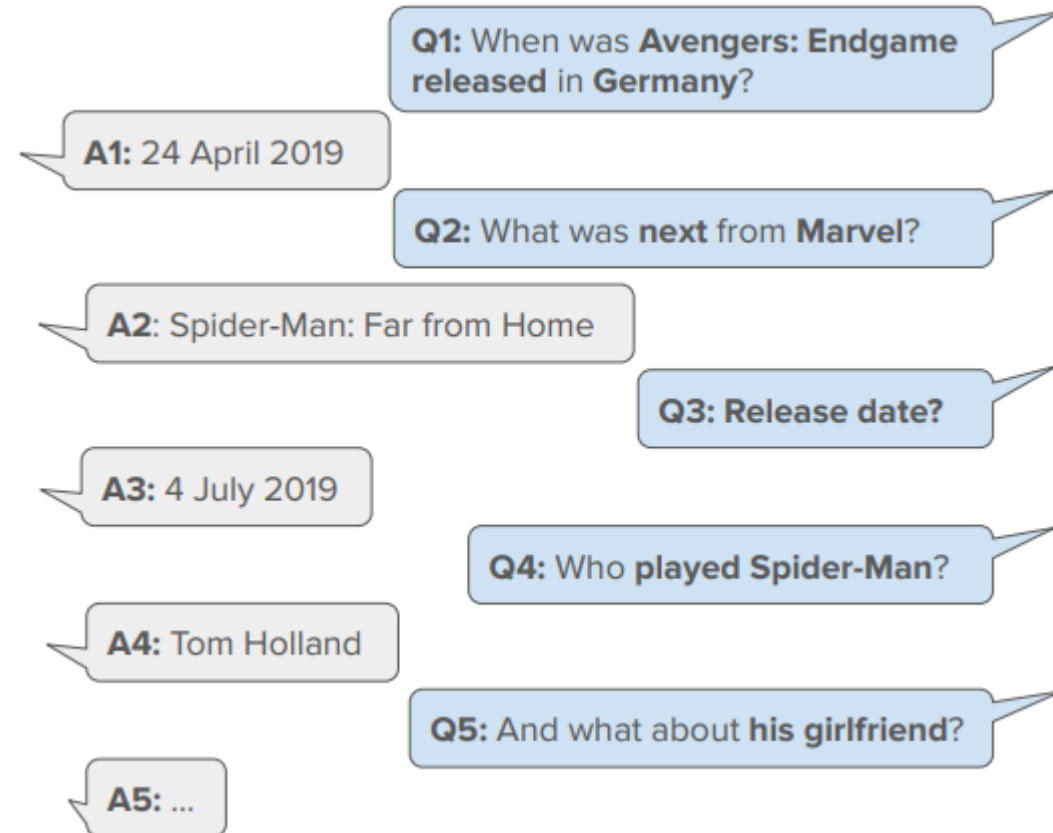


ConvQA over Knowledge Graphs

□ Key challenges

- Short, incomplete questions
- Implicit context

An example



- [1] M. Kaiser, R. Saha Roy, and G. Weikum. 2021. Reinforcement Learning from Reformulations in Conversational Question Answering over Knowledge Graphs (SIGIR 21)

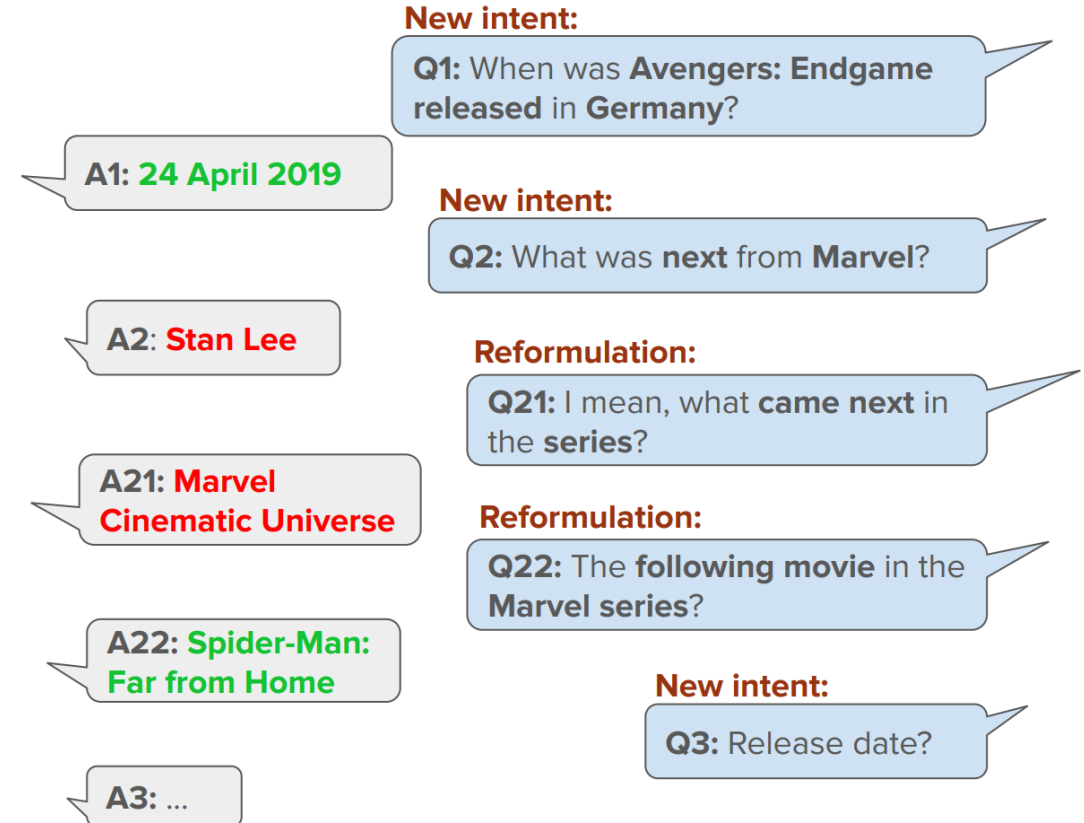
Realistic Conversation

❑ Challenges

- Short, incomplete questions
- Implicit context

❑ Learn from conversational stream

- Reformulation = Wrong answer
- New intent = Correct answer



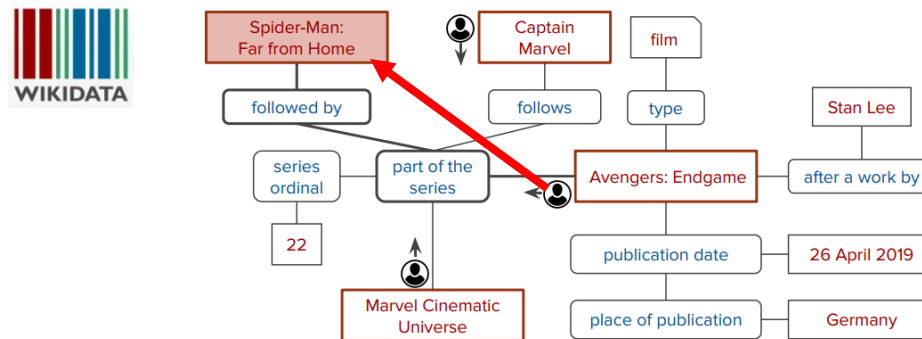
Conquer: Key Ideas

□ Key ideas: Reinforcement learning

- Multiple agents
- Rewards based on implicit feedback in form of question reformulations
- Reformulation predictor: classify the follow-up question as a reformulation or new intent

□ Architecture

Q1: When was **Avengers: Endgame** released in Germany?
 A1: 24 April 2019
 Q2: What was next from **Marvel**?

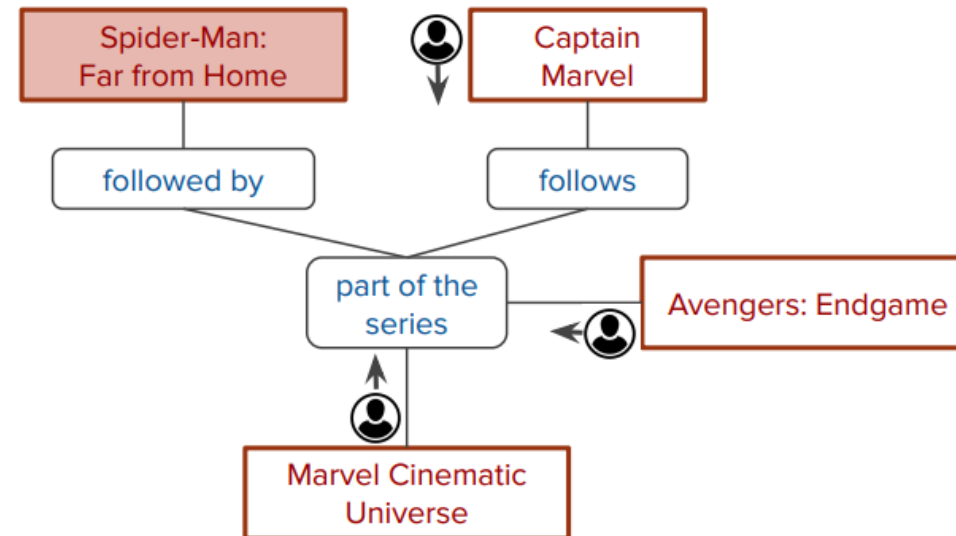


- [1] M. Kaiser, R. Saha Roy, and G. Weikum. 2021. Reinforcement Learning from Reformulations in Conversational Question Answering over Knowledge Graphs (SIGIR 21)

Conquer: Context Entity Detection

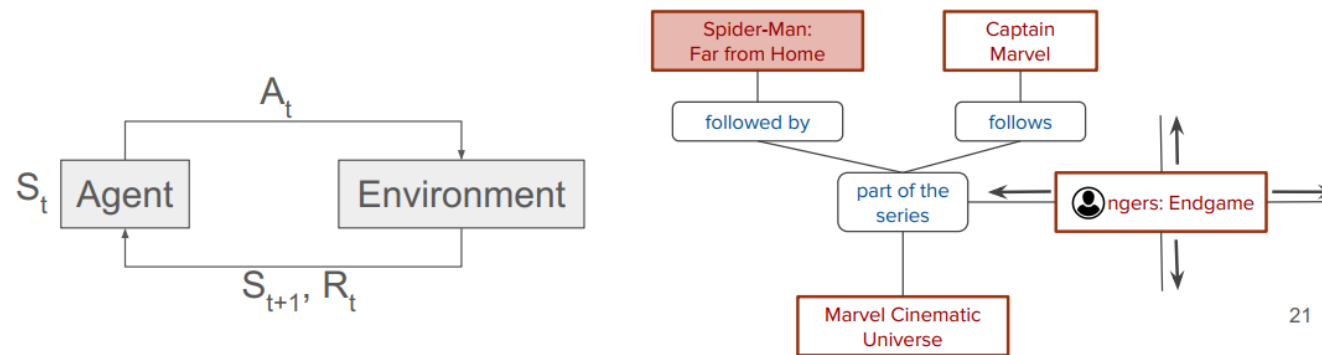
- ❑ Goal: Find entities relevant to current question and its conversational context
- ❑ Context entities will be start-points for the RL walk
- ❑ Get initial entities from first complete question via NED tool

Q1: When was **Avengers: Endgame** released in Germany?
 A1: 24 April 2019
 Q2: What was next from **Marvel**?



Conquer: Path Prediction

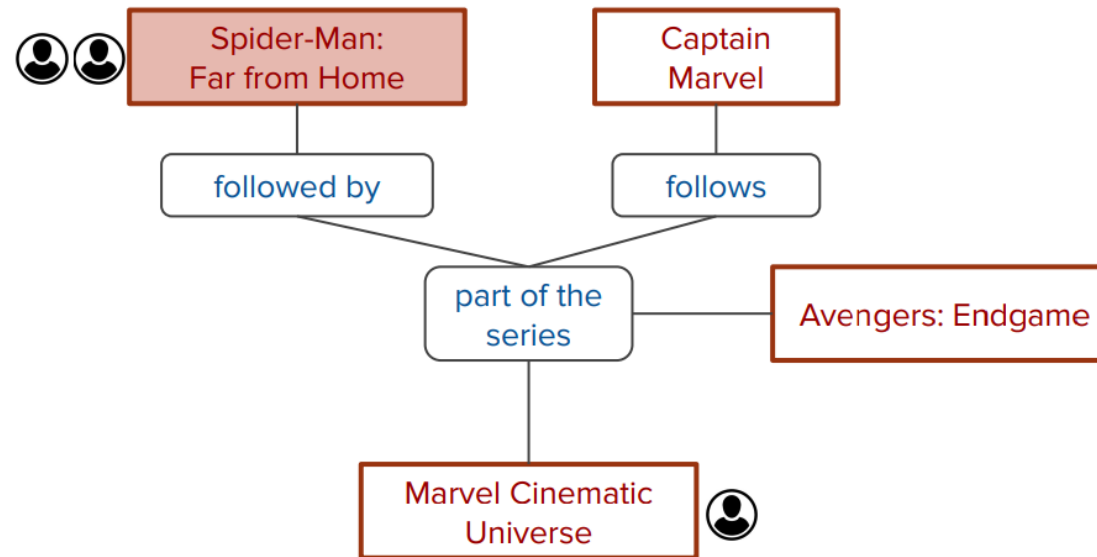
- ❑ Goal: Predict for each agent which path it should take
- ❑ Idea: MDP
 - States: current question, context entity, conversation history (optional)
 - Actions: all outgoing paths from the context entity node
 - Rewards: 1 if next question is a **new intent**, -1 if it is a **reformulation**



21

Conquer: Answer Generation

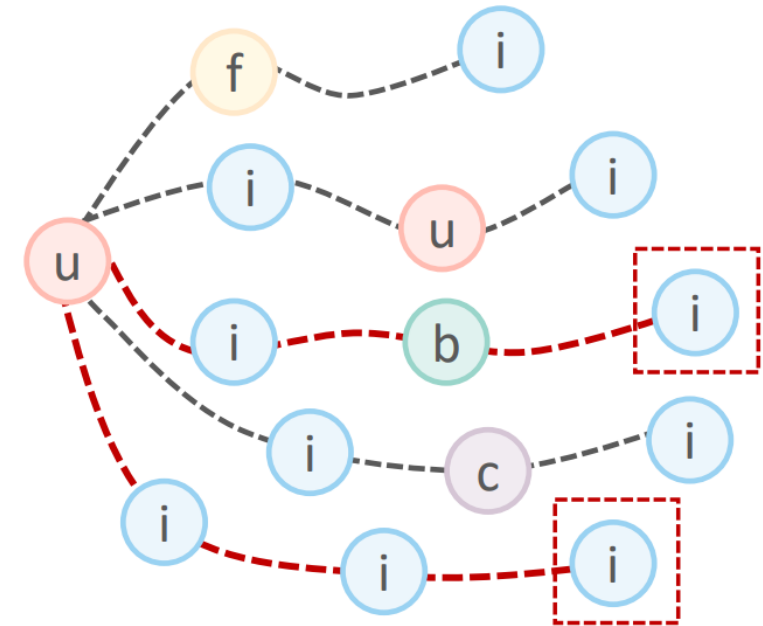
- ❑ Take top actions and rank them
- ❑ Main ranking criterion: prediction score from policy network, boosted if several agents arrive at same answer entity



- [1] M. Kaiser, R. Saha Roy, and G. Weikum. 2021. Reinforcement Learning from Reformulations in Conversational Question Answering over Knowledge Graphs (SIGIR 21)

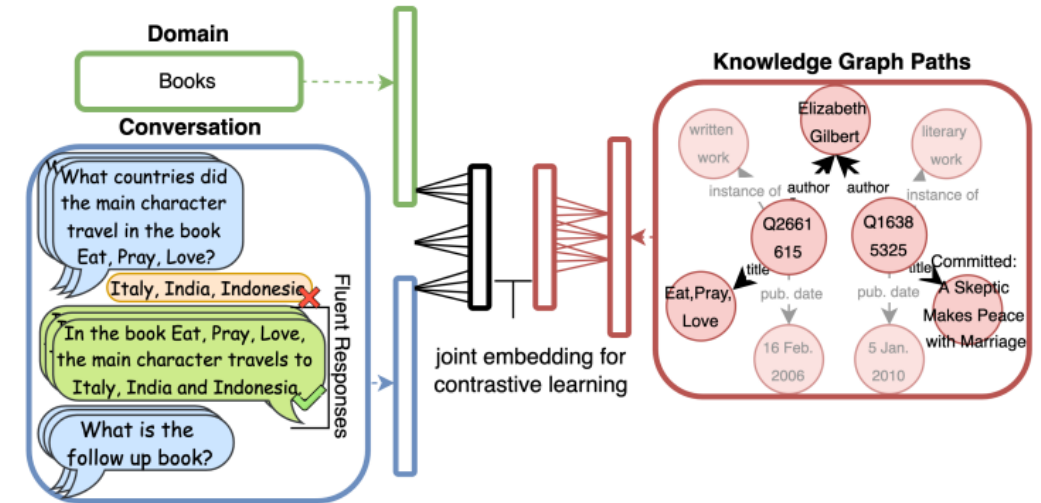
Disadvantages of RL-based ConvQA Model

- ❑ Paths are very similar, **hard to distinguish**
- ❑ Reward is **sparse**
 - Hard to train



PRALINE: Key Ideas

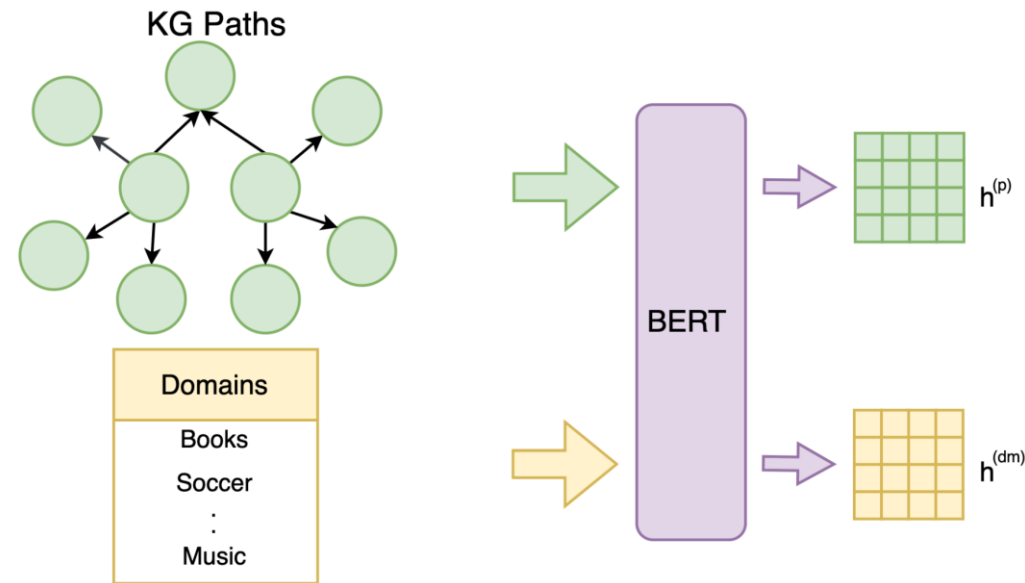
- ❑ Contrastive representation learning approach to rank KG paths for retrieving the correct answers effectively
 - Positive: paths leading to correct answers
- ❑ Enrich the learning process by incorporating
 - The entire dialog history
 - Verbalized answers
 - Domain information



Step #1: Preprocessing

- ❑ Given the conversation history and current question, extracting candidate paths
- ❑ Treat each path as a sentence, learn path embedding by BERT

1. Preprocessing / Extracting Paths & Representations



- [1] E. Kacupaj, K. Singh, M. Maleshkova, J. Lehmann. 2022. Contrastive Representation Learning for Conversational Question Answering over Knowledge Graphs (CIKM 22)

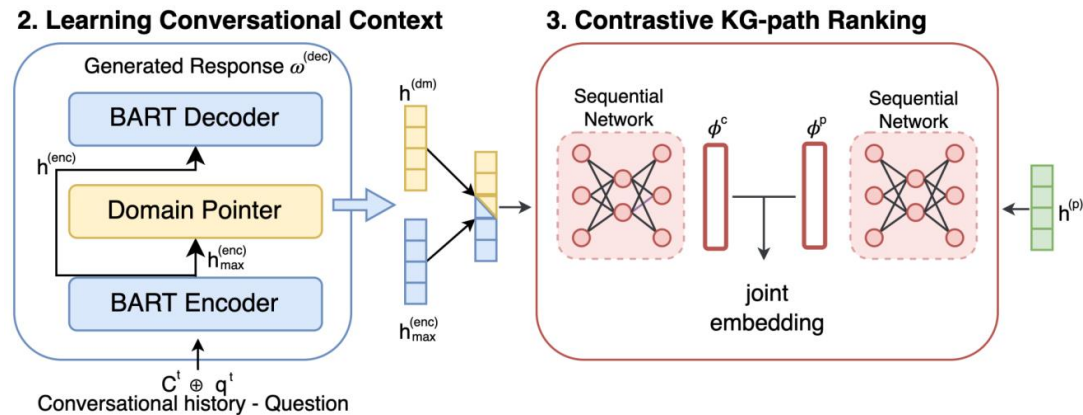
Step #2: Learning Conversational Context Information

□ Context embedding

- Utilize a BART model to encode both the conversation history and question at each turn

□ Verbalized answer

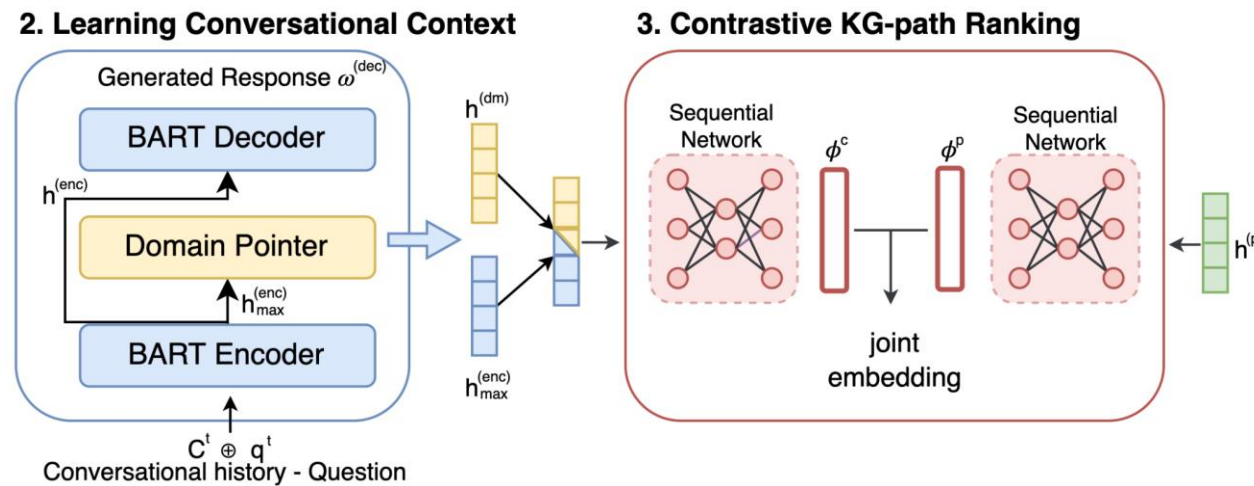
- Using a BART model to generate the verbalized answer for each entity answer



- [1] E. Kacupaj, K. Singh, M. Maleshkova, J. Lehmann. 2022. Contrastive Representation Learning for Conversational Question Answering over Knowledge Graphs (CIKM 22)

Step #3: Contrastive KG-path Ranking

- Given a batch of (conversational context, KG paths) pairs during training
 - Learn **conversation context embedding + domain embedding** by Step #2
 - Learn **path embedding** by Step #1
 - Maximize the similarity between the correct pairs while minimizing the similarity for incorrect pairs



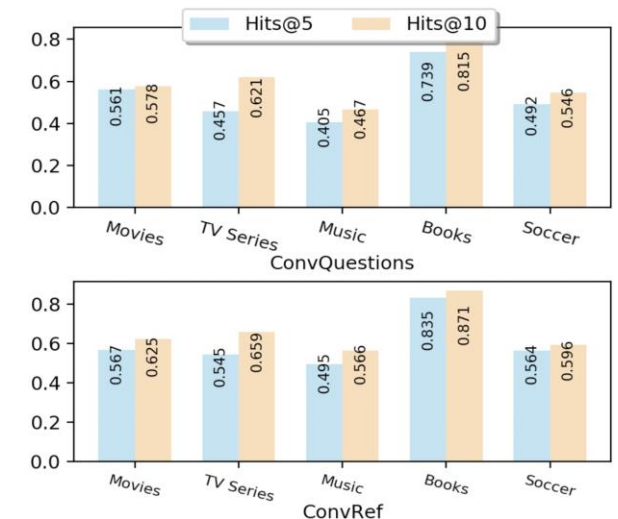
- [1] E. Kacupaj, K. Singh, M. Maleshkova, J. Lehmann. 2022. Contrastive Representation Learning for Conversational Question Answering over Knowledge Graphs (CIKM 22)

PRALINE: Experiment

□ Observations

- Evaluate PRALINE on two relevant datasets: ConvQuestions and ConvRef
- Outperform all other baselines
- Good performance on different domains

Dataset	ConvQues.			ConvRef		
Model	P@1	H@5	MRR	P@1	H@5	MRR
CONVEX [10]	0.184	0.219	0.200	0.225	0.257	0.241
CONQUER [20]	0.240	0.343	0.279	0.353	0.429	0.387
OAT [28]	0.250	-	0.260	-	-	-
Focal Entity [23]	0.248	-	0.248	-	-	-
PRALINE	0.292	0.529	0.398	0.335	0.599	0.441



H@5 and H@10 ranking results.

Roadmap

Part I: Knowledge Graph Reasoning: Basic Concepts and Techniques

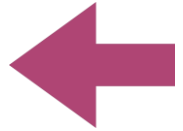
Part II: Recent Advance #1: Neural Reasoning for Natural Language Queries

Part III: Recent Advance #2: Neural Reasoning for Logical Queries

Part IV: Recent Advance #3: Neural Reasoning Beyond Entities and Relations

Part V: Recent Advance #4: LLM+KGR

Part VI: Open Challenges and Future Directions



New Frontiers of Knowledge Graph Reasoning: Recent Advances and Future Trends

Part III, Neural Reasoning for Logical Queries

Zihao Wang

Department of CSE, HKUST



KnowComp Group
Understanding the World by Computational Knowledge

Roadmap

Part I: Knowledge Graph Reasoning: Basic Concepts and Techniques

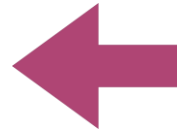
Part II: Recent Advance #1: Neural Reasoning for Natural Language Queries

Part III: Recent Advance #2: Neural Reasoning for Logical Queries

Part IV: Recent Advance #3: Neural Reasoning Beyond Entities and Relations

Part V: Recent Advance #4: LLM+KGR

Part VI: Open Challenges and Future Directions



Why study logical queries

For natural language queries, a neural model should

- Understanding natural language.
- Producing desired answers with knowledge graphs.

For logical queries, a neural model focuses only on

- Reasoning with knowledge graphs.
- Connection: A simplified version where natural language is fully understood.

Natural Language:

Find **non-American** **directors** whose movies **won** Golden Globes or Oscars.

Logical Query:

$$q = y \exists x_1. (\text{Won}(x_1, \text{GoldenGlobes}) \vee \text{Won}(x_1, \text{Oscar})) \wedge \neg \text{BornIn}(y, \text{America}) \wedge \text{Direct}(y, x_1)$$

Problem setup:

- A logical query, usually in Existential First Order (EFO) (well-studied in the database literature).
- An observed $\mathcal{KG}_o$, could be **incomplete** and noisy. (more complex KGs will be discussed later)
- An enriched $\mathcal{KG}_e$, could be completed by link prediction $(h, r, ?)$

Background: Query syntax

Natural Language:

Find **non-American directors** whose movies **won** Golden Globes **or** Oscars.

Logical Query:

$$q = y \exists x_1. (\text{Won}(x_1, \text{GoldenGlobes}) \vee \text{Won}(x_1, \text{Oscar})) \wedge \neg \text{BornIn}(y, \text{America}) \wedge \text{Direct}(y, x_1)$$

Syntax of Existential First Order (EFO) query family
(organized as Unions of Conjunctive Queries (UCQ))

- An UCQ query is represented as the disjunction of conjunctive queries,

$$UCQ(y; x_1, \dots, x_n) = \bigvee_{j=1, \dots, N} \boxed{CQ_j(y; x_1, \dots, x_n)}$$

y is the variable of our interest.

x_i are existential variables that we don't need to find out.

$$q(y; x) = \text{SetUnion} \left(\begin{array}{l} y \exists x_1. (\text{Won}(x_1, \text{GoldenGlobes}) \wedge \neg \text{BornIn}(y, \text{America}) \wedge \text{Direct}(y, x_1)), \\ y \exists x_1. (\text{Won}(x_1, \text{Oscar}) \wedge \neg \text{BornIn}(y, \text{America}) \wedge \text{Direct}(y, x_1)) \end{array} \right)$$

Background: Query syntax

Syntax of Existential First Order (EFO) query family
(organized as Unions of Conjunctive Queries (UCQ))

- An UCQ query is represented as the disjunction of conjunctive queries,

$$UCQ(y; x_1, \dots, x_n) = \bigvee_{j=1, \dots, N} CQ_j(y; x_1, \dots, x_n)$$

- Each conjunctive query is the conjunctive of atomic formulas,

$$CQ_j(y; x_1, \dots, x_n) = y. \exists x_1, \dots, \exists x_n. \bigwedge_{k=1, \dots, M_j} a_{jk}$$

$$y \exists x_1. (\text{Won}(x_1, \text{GoldenGlobes}) \wedge \neg \text{BornIn}(y, \text{America}) \wedge \text{Direct}(y, x_1))$$

- Each atomic formula is $a_{jk} = r(t_s, t_o)$, or $a_{jk} = \neg r(t_s, t_o)$,
 - where r is the binary relation in KG,
 - t_s and t_o are the subjective/objective terms, respectively,
 - Each term is either an entity or variable $(y, x_1, \dots, x_n)$.

Background: Query Semantics

Answer set

- $A = \{a \in \mathcal{E} : Q(y = a; x_1, \dots, x_n) = \text{True}\}.$
- Depends on the semantics of the substitution $Q(\mathbf{y} = \mathbf{a}; x_1, \dots, x_n).$

$$\begin{aligned} q(y; x_1) &= y \exists x_1. (\text{Won}(x_1, \text{GoldenGlobes}) \vee \text{Won}(x_1, \text{Oscar})) \wedge \neg \text{BornIn}(y, \text{America}) \wedge \text{Direct}(y, x_1) \\ q(\mathbf{y} = \mathbf{a}; x_1) &= \exists x_1. (\text{Won}(x_1, \text{GoldenGlobes}) \vee \text{Won}(x_1, \text{Oscar})) \wedge \neg \text{BornIn}(\mathbf{y} = \mathbf{a}, \text{America}) \wedge \text{Direct}(\mathbf{y} = \mathbf{a}, x_1) \end{aligned}$$

Q: How to evaluate $Q(\mathbf{y} = \mathbf{a}; x_1, \dots, x_n)?$

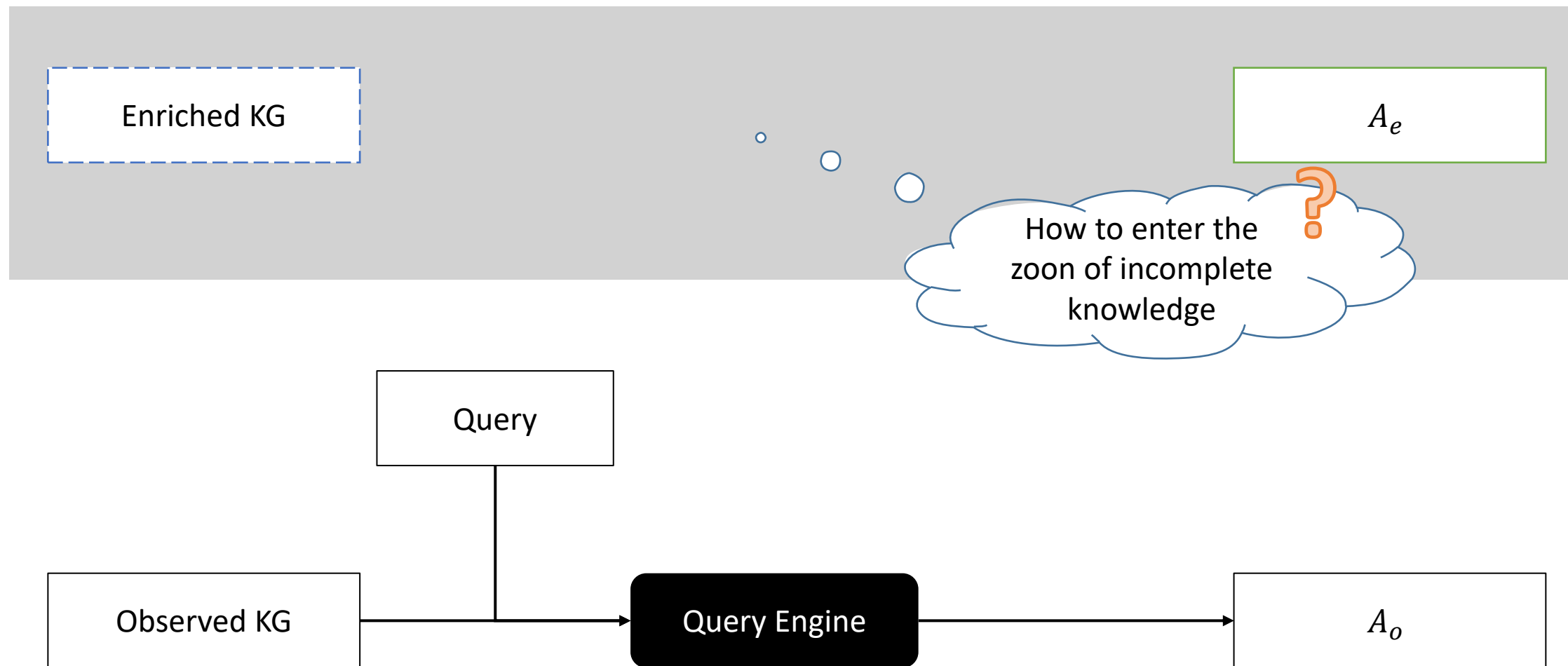
A: Evaluate the expansion, reduce to the model checking problem

$$\bigvee_{j=1, \dots, N} \exists x_1, \dots, \exists x_n. \bigwedge_{k=1, \dots, M_j} a_{jk} \Big|_{\mathbf{y}=\mathbf{a}}$$

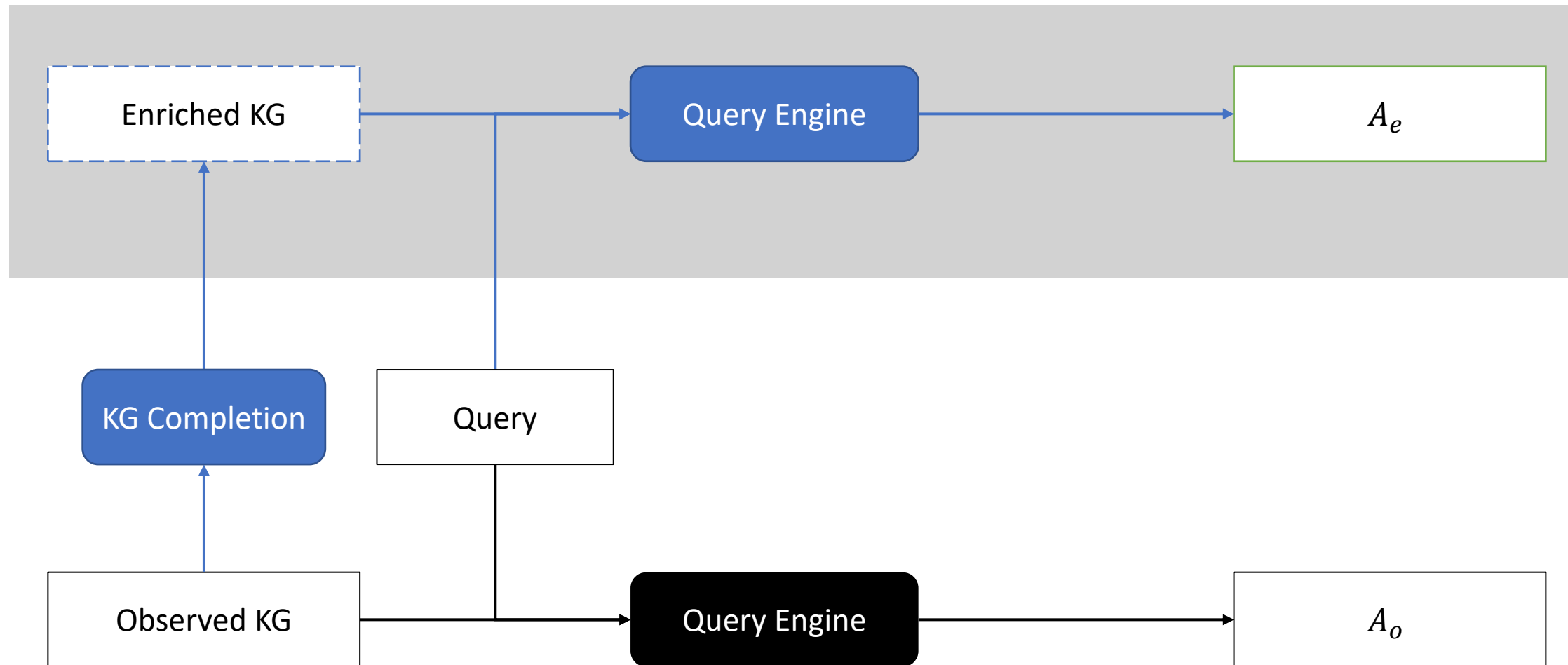
Then, it eventually depends on each atomic formula a_{jk} .

- In the database setting (Answer set A_o):
 - $r(s, o) = \text{True}$ if and only if $(s, r, o) \in \mathcal{KG}_o$, which is traversal on observed KG.
- In our setting, with enriched knowledge (Answer set A_e):
 - $r(s, o) = \text{True}$ if and only if $(s, r, o) \in \mathcal{KG}_e$, where $\mathcal{KG}_e$ is enriched.

The setting of our interest



Strategy I: completion and search



Fuzzy Inference with Truth values

Calculate the estimated truth value **TV** defined with fuzzy logic.

Tools for estimation:

- A link predictor $f(h, r, t) \in [0,1]$, evaluating atomic formulae $TV[a] = TV(r(h, t))$
- A t -norm $\top$ for conjunction
 - $TV[q_1(y = a) \wedge q_2(y = a)] = TV[q_1(y = a)] \top TV[q_2(y = a)]$
- A t -conorm $\perp$ for disjunction (paired with $\top$)
 - $TV[q_1(y = a) \vee q_2(y = a)] = TV[q_1(y = a)] \perp TV[q_2(y = a)]$
- A t -conorm $\perp^*$ for existential variable
- A logical negator $TV[\neg a] = 1 - TV[a]$

Godel t -norm

- $a \top b = \min(a, b)$
- $a \perp b = \max(a, b)$

Product norm

- $a \top b = ab$
- $a \perp b = a + b - ab$

Fuzzy Inference with Truth values

Then, the evaluation problem goes

$$\begin{aligned}
 & \textcolor{red}{TV}[Q(\textcolor{blue}{y} = \textcolor{blue}{a}; x_1, \dots, x_n)] \\
 &= \textcolor{red}{TV} \left[\bigvee_{j=1, \dots, N} \exists x_1, \dots, \exists x_n. \bigwedge_{k=1, \dots, M_j} a_{jk} \Big|_{\textcolor{blue}{y}=\textcolor{blue}{a}} \right] \\
 &= \bigwedge_{j=1, \dots, N} \left(\perp_{x_1=e_1 \in \mathcal{E}}^* \cdots \perp_{x_n=e_n \in \mathcal{E}}^* \bigvee_{k=1, \dots, M_j} \textcolor{red}{TV} \left[a_{jk} \Big|_{\textcolor{blue}{y}=\textcolor{blue}{a}} \right] \right)
 \end{aligned}$$

Fuzzy Inference with Truth values

- When $\perp^*$ is a Godel t -conorm, the original inference problem

$$TV[Q(y = a)] = \perp_{x_1=e_1 \in \mathcal{E}}^* \cdots \perp_{x_n=e_n \in \mathcal{E}}^* \bigwedge_{j=1, \dots, N} \bigvee_{k=1, \dots, M_j} TV \left[a_{jk} \middle|_{y=a} \right]$$

becomes a search problem

$$TV[Q(y = a)] = \max_{x_1, \dots, x_n \in \mathcal{E}} \bigwedge_{j=1, \dots, N} \bigvee_{k=1, \dots, M_j} TV \left[a_{jk} \middle|_{y=a} \right]$$

- The complexity of this problem in general grows exponentially with respect to the number of variables
 - More discussion can be found in Yin et al, ICLR'24

A simpler case of the inference problem

Remarks

- The search problem can be drastically simplified when the query graph (nodes are variables, and edges are atomics) is a tree.
- each existential variable is removed by $O(|\mathcal{E}|^2)$.
- The overall complexity is linear to the number of variables.
- Search on DAGs is by Bai et al., ICML'23
- The beam search version on DAGs is by Arakelyan et al., ICLR'21

Search in the continuous space

The search problem

$$TV[Q(y = a)] = \max_{x_1, \dots, x_n \in \mathcal{E}} \perp_{j=1, \dots, N} \top_{k=1, \dots, M_j} TV \left[a_{jk} \Big|_{y=a} \right]$$

is defined over the discrete set $\mathcal{E}$.

A continuous relaxation put $x_1, \dots, x_n$ in the embedding space $\mathcal{X}$

$$TV[Q(y = a)] = \max_{x_1, \dots, x_n \in \mathcal{X}} \perp_{j=1, \dots, N} \top_{k=1, \dots, M_j} TV \left[a_{jk} \Big|_{y=a} \right]$$

- The optimization objective is differentiable because
 - $\top$, $\perp$, and $TV \left[a_{jk} \Big|_{y=a} \right]$ are differentiable.

- This problem can be solved via **gradient ascend**

- A quick version to get optimal $x_1, \dots, x_n$

$$\max_{x_1, \dots, x_n \in \mathcal{X}, y \in \mathcal{X}} \perp_{j=1, \dots, N} \top_{k=1, \dots, M_j} TV[a_{jk}]$$

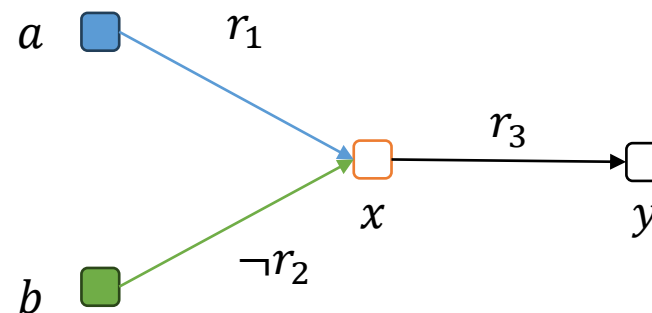
Learning to optimize in the continuous space

Can we skip the gradient ascend?

- Consider the **conjunctive queries**

$$\max_{x_1, \dots, x_n \in \mathcal{X}, y \in \mathcal{Y}} \prod_{k=1, \dots, M_j} TV[a_{jk}]$$

- This conjunctive queries can be considered as a query graph
 - Logical form: $y \exists x. r_1(a, x) \wedge \neg r_2(b, x) \wedge r_3(x, y)$
 - Optimization form: $\max_{x, y} f(a, r_1, x) \prod [1 - f(b, r_2, x)] \prod f(x, r_3, y)$
 - Or the graph in the right.



Learning to optimize in the continuous space

Optimize

Goal: optimize the embedding of x, y

Method: gradient ascend

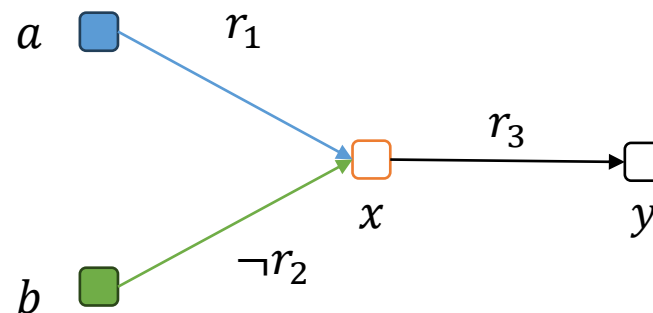
$$\max_{x,y} f(a, r_1, x) \top [1 - f(b, r_2, x)] \top f(x, r_3, y)$$

Learning to optimize

Goal: estimate the embedding of x, y

Method: neural network forward pass

New design problem of NN akin to the optimization process



Learning to optimize in the continuous space

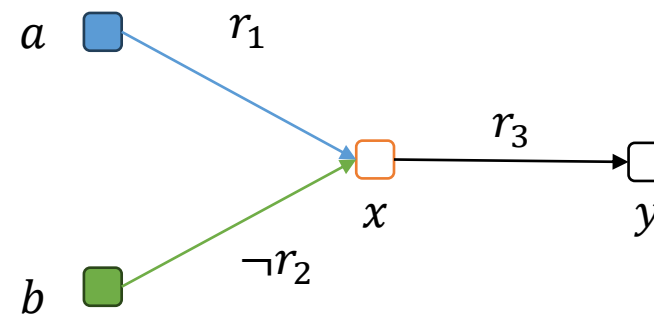
Optimize the *local objective* in each edge (atomic formula) with closed-form solutions, instead of the *global objective*:

$$\max_{x,y} f(a, r_1, x) \top [1 - f(b, r_2, x)] \top f(x, r_3, y)$$

- **One-hop** inference problems

- $\rho(h, r, h2t, 0) := \max_x f(h, r, x)$
- $\rho(t, r, t2h, 0) := \max_x f(x, r, t)$
- $\rho(h, r, h2t, 1) := \max_x 1 - f(h, r, x)$
- $\rho(t, r, t2h, 1) := \max_x 1 - f(x, r, t)$
- $\rho(\text{entity}, \text{relation}, \text{direction}, \text{negation})$ is a message function

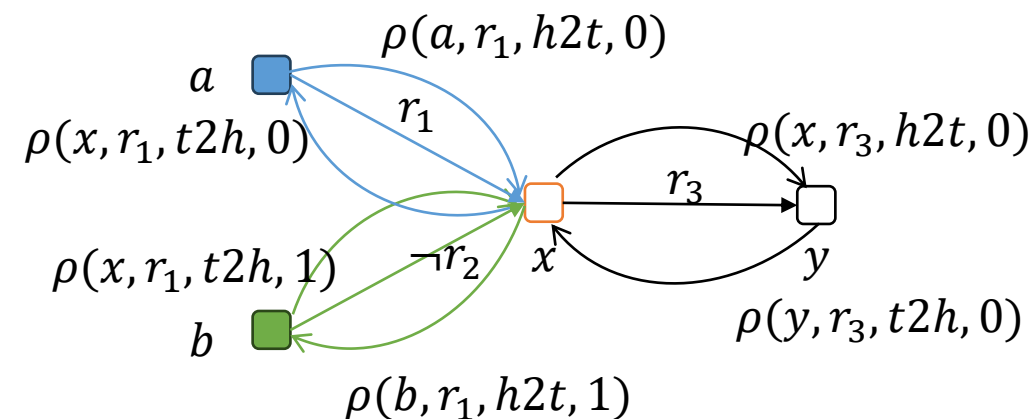
- Closed-form ρ can be found for many link predictors.



Learning to search in the continuous space

The GNN design: Logical Message Passing Neural Networks

- GNN layer: Message function ρ + MLP update
 - Number of GNN layers = diameter of the query graph
 - Reuse the same MLP in all layers
 - Able to match various query size
 - Initialization
 - **Entities: pretrained KGE**
 - **Existential variables: a special token**
 - **Free variable: another special token**
- ✓ The embed. of x, y are estimated by passing GNN layers.

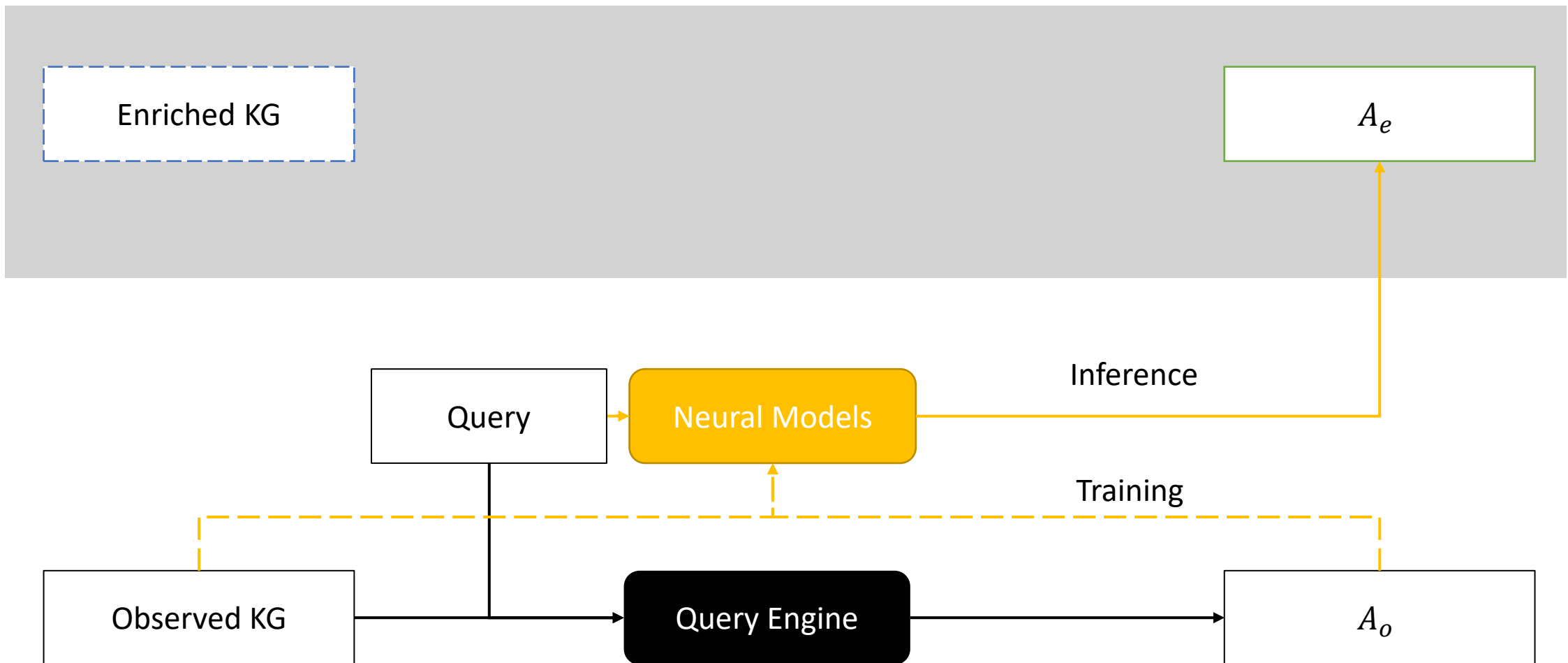


GNN end-to-end Training

Loss function: noisy contrastive estimation:

$$L = -\log \frac{e^{\cos(y,a)}}{e^{\cos(y,a)} + \sum_k e^{\cos(y,e_k^-)}}$$

Strategy II: end-to-end training



What to neutralize?

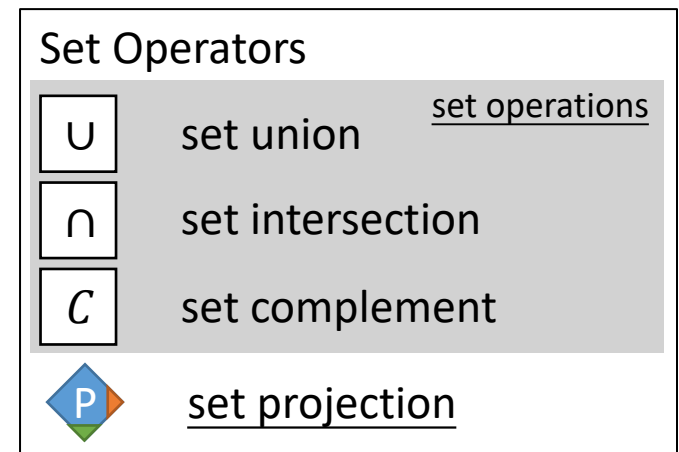
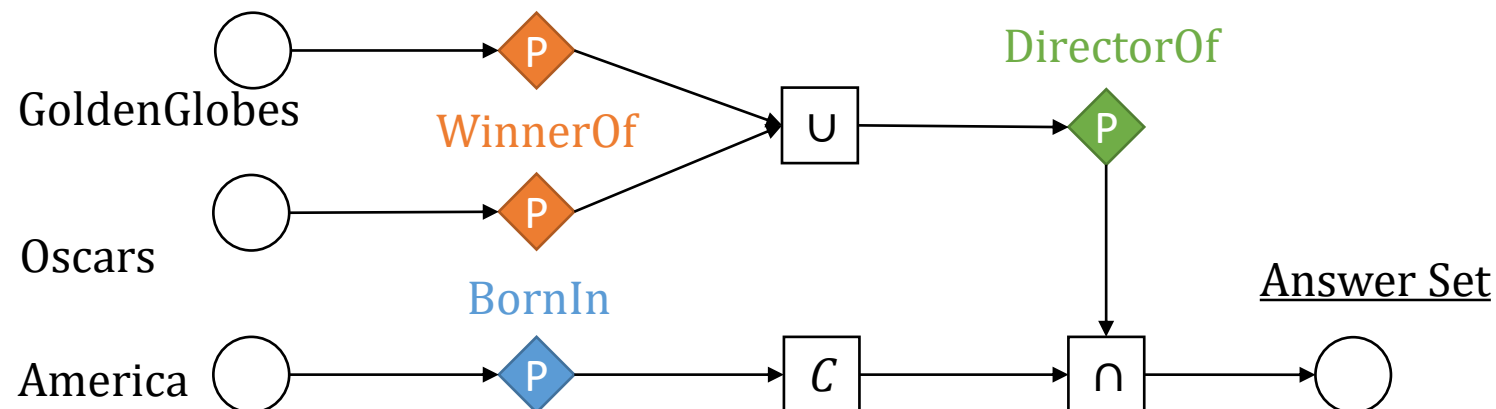
Presentation	Model type	Example	Logic Modules
Linearization into string	Sequence models	Transformer	Weak
Graph of terms and predicates	Graph models	GNN	Mid
Process of execution	<u>Neural Set Operators</u>	<u>Query Embedding</u>	Strong

Natural Language:

Find **non-American** **directors** whose movies **won** Golden Globes or Oscars.

Logical Query:

$$q = y \exists x_1. (\text{Won}(x_1, \text{GoldenGlobes}) \vee \text{Won}(x_1, \text{Oscard})) \wedge \neg \text{BornIn}(y, \text{America}) \wedge \text{Direct}(y, x_1)$$



The design space of neural set operators

Concept	Definition	Comment
Entity set	$\mathcal{E}$	The entity set in KG
Relation set	$\mathcal{R}$	The relation set in KG
Set embedding space	$\mathcal{X}$	Embedding space [To design]
Set embedding lookup	$E_{\mathcal{X}}: \mathcal{E} \mapsto \mathcal{X}$	Singleton set embedding
Entity embedding space	$\mathcal{Y}$	Embedding space [To design]
Entity embedding lookup	$E_{\mathcal{Y}}: \mathcal{E} \mapsto \mathcal{Y}$	Entity embedding
Set intersection	$I: \mathcal{X} \times \dots \times \mathcal{X} \mapsto \mathcal{X}$	Binary or N-ary [To design]
Set union	$U: \mathcal{X} \times \dots \times \mathcal{X} \mapsto \mathcal{X}$	Binary or N-ary [To design]
Set complement	$C: \mathcal{X} \mapsto \mathcal{X}$	Replaceable with set difference [To design]
Set projection	$P: \mathcal{X} \times \mathcal{R} \mapsto \mathcal{X}$	One-hop link prediction [To design]
Scoring function	$s: \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$	How much an entity is in a set [To design]

Converting to computational tree makes it possible to model **set operations** with neural networks

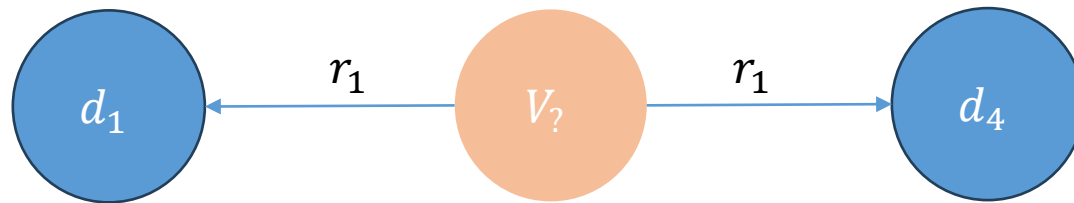
Set Operators

$\cup$	set union	<u>set operations</u>
$\cap$	set intersection	
C	set complement	
P	set projection	

An example of set operation

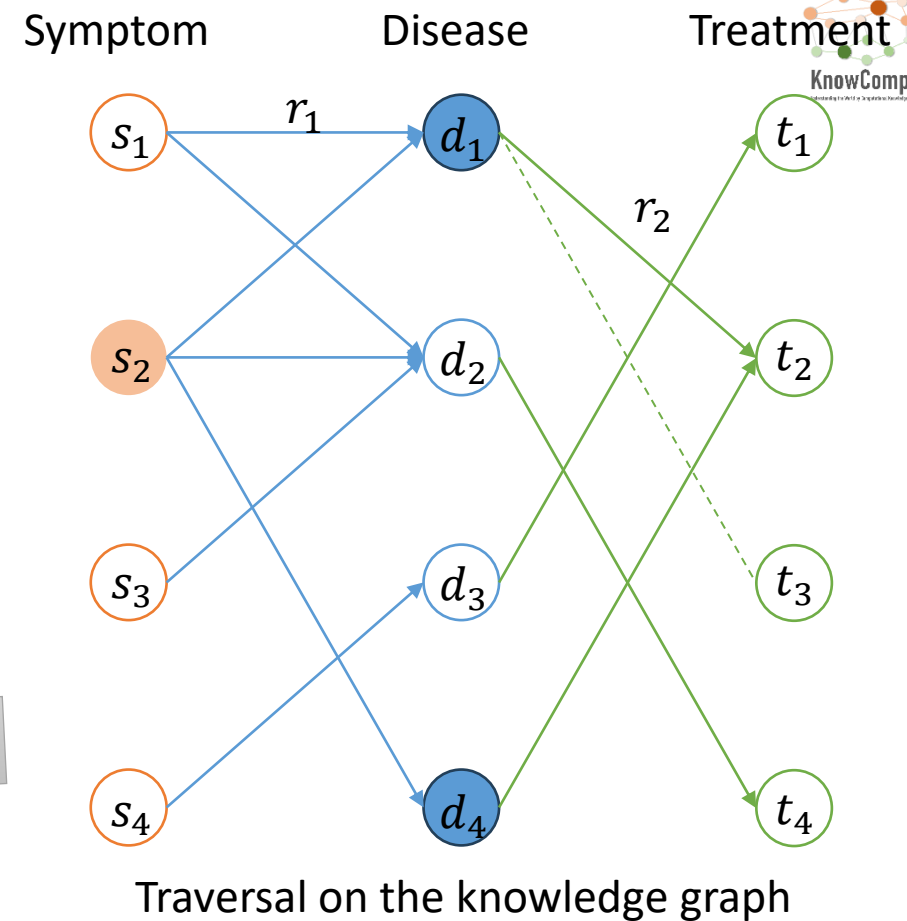
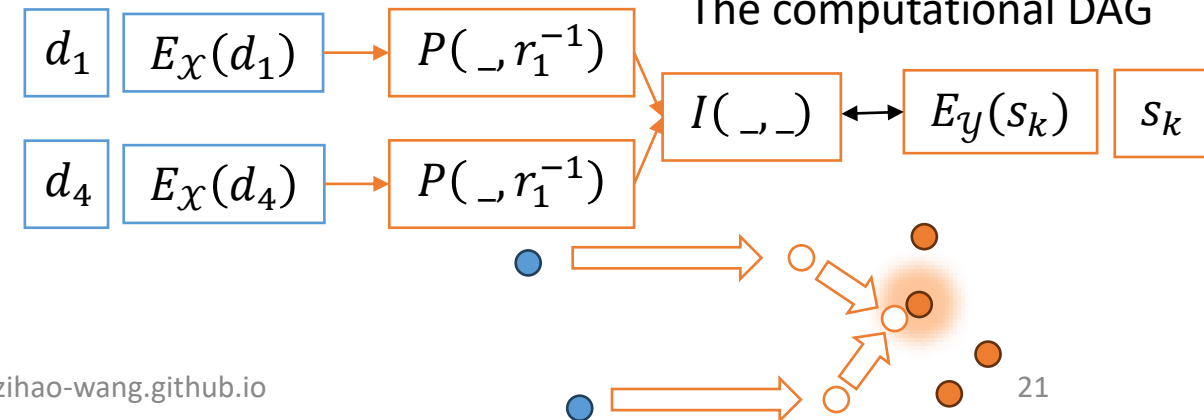
Logical query in natural language	Logical query in formal language
What is the common symptom of diseases "d ₁ " and "d ₄ "?	$V_?. r_1(V_?, d_1) \wedge r_1(V_?, d_4)$

Set Intersection



1. Computation of query embedding:

$$\hat{y} = I\left(P(E_x(d_1), r_1^{-1}), P(E_x(d_4), r_1^{-1})\right)$$
2. Rank entities by $s(\hat{y}, E_y(s_1)), s(\hat{y}, E_y(s_2)), \dots$



Examples in $\mathbb{R}^d$: Vector

GQE (Hamilton et al., 2018)

Definition	Comment
$\mathcal{X}$	$q \in \mathbb{R}^d$
$\mathcal{Y}$	$a \in \mathbb{R}^d$
$I: \mathcal{X} \times \dots \times \mathcal{X} \mapsto \mathcal{X}$	$I(q_1, \dots, q_n) = W\Psi(\text{MLP}(q_1), \dots, \text{MLP}(q_n))$
$U: \mathcal{X} \times \dots \times \mathcal{X} \mapsto \mathcal{X}$	UCQ equivalent forms
$C: \mathcal{X} \mapsto \mathcal{X}$	NA
$P: \mathcal{X} \times \mathcal{R} \mapsto \mathcal{X}$	$P(q, r) = R_r q$
$s: \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$	$s(q, a) = \frac{q^T a}{\ q\ \ a\ }$

MLP: multi-layer perceptron

Ψ : a permutation invariant operator

W : a matrix

R_r : a matrix indexed by relation r

Neural operations do almost everything

Example in $\mathbb{R}^d$: multiple vector

Query2Particles (Bai et al., 2022)

Definition	Comment
$\mathcal{X}$	Multiple vectors in $\mathbb{R}^d$
$\mathcal{Y}$	$\mathbb{R}^d$
$I: \mathcal{X} \times \dots \times \mathcal{X} \mapsto \mathcal{X}$	$A_i = \text{self-attn}(P_i)$ $P_{i+1} = \text{MLP}(A_i)$
$U: \mathcal{X} \times \dots \times \mathcal{X} \mapsto \mathcal{X}$	Merge Particles
$C: \mathcal{X} \mapsto \mathcal{X}$	$A_i = \text{self-attn}(P_i)$ $P_{i+1} = \text{MLP}(A_i)$
$P: \mathcal{X} \times \mathcal{R} \mapsto \mathcal{X}$	$A_i = (1 - Z) \odot P_i + Z \odot T$ $P(q, r) = \text{self-attn}(A_i)$
$s: \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$	$s(q, a) = \max_{k=1,2,3,\dots,K} \langle p_T^{(k)}, a \rangle$

Because of multiple particles describe one set, then we can describe the set union in a natural way.

Meanwhile, it enables many advanced neural network architectures

Such as

- self-attention
- Gates

Gated Transition for customizing the directions of transitions for each vector in particles:

$$A_i = (1 - Z) \odot P_i + Z \odot T$$

Here Z is the update gate, and T is transition for each particles. They are computed from P_i and the relation embedding e_l for relation l

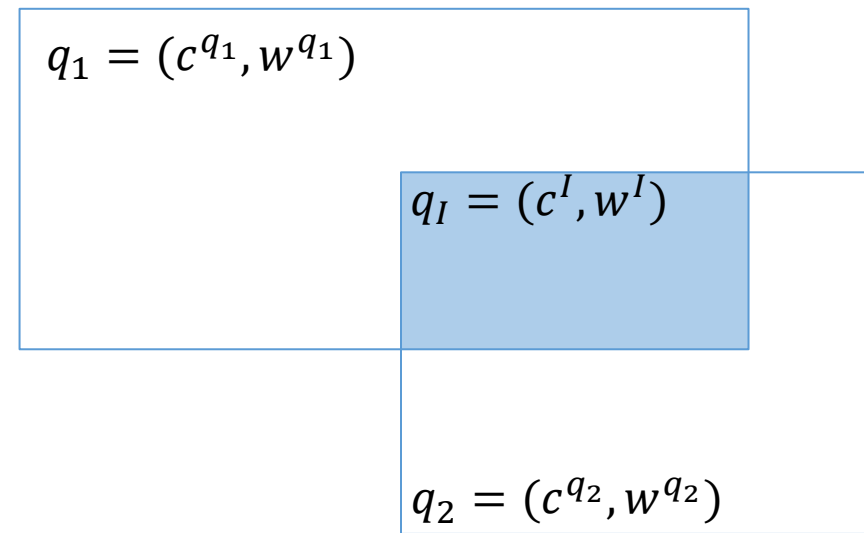
Example in Geometric regions

Query2Box (Ren et al., 2020)

Definition	Comment
$\mathcal{X}$	q is a box in $\mathbb{R}^d$
$\mathcal{Y}$	$a \in \mathbb{R}^d$
$I: \mathcal{X} \times \dots \times \mathcal{X} \mapsto \mathcal{X}$	$q_I = I(q_1, \dots, q_i, \dots, q_n)$ $c^I = \sum a_i c^{q_i}, a_i = \text{softmax}_{i=1, \dots, n}(\text{MLP}(q_i))$ $w^I = \min\{w^{q_1}, \dots, w^{q_n}\} \sigma(\text{Deepset}(q_1, \dots, q_n))$
$U: \mathcal{X} \times \dots \times \mathcal{X} \mapsto \mathcal{X}$	UCQ
$C: \mathcal{X} \mapsto \mathcal{X}$	NA
$P: \mathcal{X} \times \mathcal{R} \mapsto \mathcal{X}$	$\text{Box}(c^{P(q,r)}, w^{P(q,r)}) = \text{Box}(c_q + c_r, w_q + w_r)$
$s: \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$	$s(q, a) = \gamma - \text{dist}_{\text{outside}}(q, a) - \alpha \text{dist}_{\text{inside}}(q, a)$

Training objective $L(q) = -\log \sigma(s(q, a)) - \sum_{j=1, \dots, k} \frac{1}{k} \log \sigma(s(q, e_j^-))$

Natural abstraction of set intersection



Example in Probabilistic distributions

BetaE (Ren and Leskovec, 2020)

Definition	Comment
$\mathcal{X}$	$q = (\alpha_1^q, \beta_1^q, \dots, \alpha_d^q, \beta_d^q) \in [0, \infty)^{2d}$
$\mathcal{Y}$	$a \in [0, \infty)^{2d}$
$I: \mathcal{X} \times \dots \times \mathcal{X} \mapsto \mathcal{X}$	$I(q_1, \dots, q_i, \dots, q_n) = \sum_i \alpha_i q_i,$ $\alpha_i = \text{softmax}(\text{NN}(q_i))$
$U: \mathcal{X} \times \dots \times \mathcal{X} \mapsto \mathcal{X}$	UCQ
$C: \mathcal{X} \mapsto \mathcal{X}$	$C(q) = \mathbf{1}/q$
$P: \mathcal{X} \times \mathcal{R} \mapsto \mathcal{X}$	$P(q, r) = \text{MLP}_r(q)$
$s: \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$	$s(q, a) = \gamma - \sum_{i=1, \dots, d} \text{KL}(B(\alpha_i^q, \beta_i^q) B(\alpha_i^a, \beta_i^a))$

Q: How to model the set complement?

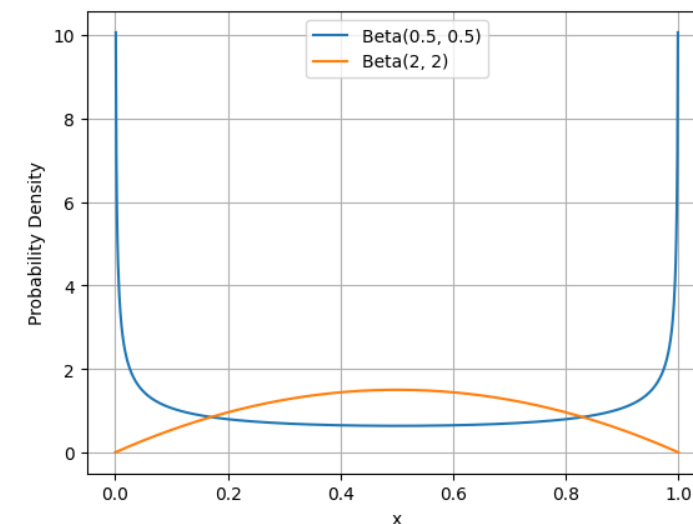
A: Use inductive bias of probability families

- P.d.f. of Beta distribution $\text{Beta}(\alpha, \beta)$

$$f(x; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1}$$

where $\Gamma(x)$ is the Gamma function.

- Set embedding: d Beta distributions.



Can “embeddings” also be logical aware?

- A matrix $M_{q,a} = TV(q(y = a))$ records everything we need.
where q is an arbitrary query and a is an entity.

$$s(q, a) = M_{q,a}$$

- $M_{q,a}$ looks large
 - M^{atomic} for finite positive atomic queries $(h, r, ?)$ are necessary
 - Other rows can be generated by fuzzy logic inference (we’ve introduced in FIT)
- Low rank decomposition of M^{atomic} .

$$M_{q,a}^{atomic} \approx \vec{q}^T \vec{a}$$

- $\vec{q}^T \in \mathbb{R}^d$ is the query embedding of an atomic query $q(y) = r(e, y)$.
- Any atomic query can be written as $q(y) = r(e, y)$ by allowing reverse relation.

Matrix decomp. leads to element-wise fuzzy logic ops

- Low rank (rank d) decomposition of M^{atomic}

$$M_{q,a}^{atomic} \approx \vec{q}^T \vec{a}$$

- $\vec{q}$ is the query embedding
- $\vec{a}$ is the entity embedding
- If we further assume that
 - $\vec{q} \in [0,1]^d, \vec{a} \in [0,1]^d$
 - t -norm is linear, for convenience we consider Godel t -norm
- Let q_1, q_2 be atomic query, and godel norm, then

$$M_{q_1 \wedge q_2, a} = \overrightarrow{q_1 \wedge q_2}^T \vec{a} = M_{q_1, a} \top M_{q_2, a} = \min(M_{q_1, a}, M_{q_2, a}) = \min(\overrightarrow{q_1}^T \vec{a}, \overrightarrow{q_2}^T \vec{a}) = \min(\overrightarrow{q_1}^T, \overrightarrow{q_2}^T) \vec{a}$$
- Conclusion:

$$\begin{aligned} \overrightarrow{q_1 \wedge q_2} &= \min(\overrightarrow{q_1}, \overrightarrow{q_2}) = \overrightarrow{q_1} \top \overrightarrow{q_2} \\ \overrightarrow{q_1 \vee q_2} &= \max(\overrightarrow{q_1}, \overrightarrow{q_2}) = \overrightarrow{q_1} \perp \overrightarrow{q_2} \\ \neg \overrightarrow{q_1} &= 1 - \overrightarrow{q_1} \end{aligned}$$

From vector to Fuzzy vector

FuzzQE (Chen et al., 2022)

Definition	Comment
$\mathcal{X}$	$q \in [0,1]^d$
$\mathcal{Y}$	$a \in [0,1]^d$
$I: \mathcal{X} \times \dots \times \mathcal{X} \mapsto \mathcal{X}$	$I(q_1, \dots, q_i, \dots, q_n) = q_1 \top \dots \top q_n$
$U: \mathcal{X} \times \dots \times \mathcal{X} \mapsto \mathcal{X}$	$U(q_1, \dots, q_i, \dots, q_n) = q_1 \perp \dots \perp q_n$
$C: \mathcal{X} \mapsto \mathcal{X}$	$C(q) = 1 - q$
$P: \mathcal{X} \times \mathcal{R} \mapsto \mathcal{X}$	$P(q, r) = \text{MLP}_r(q)$
$s: \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$	$s(q, a) = q^T a$

T-norms are all you need

From fuzzy vectors back to geometric region

WFRE (Wang et al., 2023)

Definition	Comment
$\mathcal{X}$	$q \in [0,1]^d$
$\mathcal{Y}$	$a \in [0,1]^d$

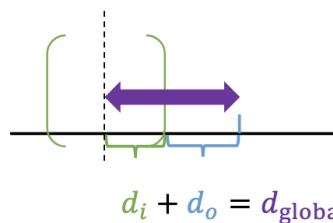
$I: \mathcal{X} \times \dots \times \mathcal{X} \mapsto \mathcal{X}$	$I(q_1, \dots, q_i, \dots, q_n) = q_1 \top \dots \top q_n$
$U: \mathcal{X} \times \dots \times \mathcal{X} \mapsto \mathcal{X}$	$U(q_1, \dots, q_i, \dots, q_n) = q_1 \perp \dots \perp q_n$
$C: \mathcal{X} \mapsto \mathcal{X}$	$C(q) = 1 - q$
$P: \mathcal{X} \times \mathcal{R} \mapsto \mathcal{X}$	$P(q, r) = \text{MLP}_r(q)$

$$s: \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$$

$$s(q, a) = \text{ConvWFR}(q, a)$$

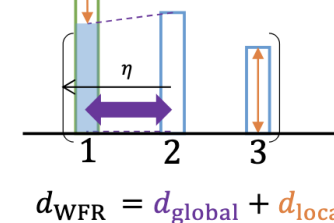
Global Transport
• Query2Box
• ConE

Region in $\mathbb{R}$



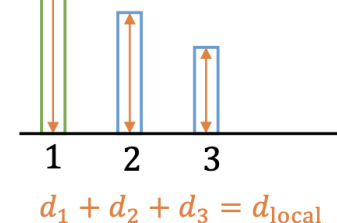
Global Transport and Local Comparison

η : transport radius



Local Comparison
• KL divergence (BetaE)
• l_p distance (MLP-MIX)

Distributions in $\mathbb{R}$



WFR (Wasserstein-Fisher-Rao) distances combines

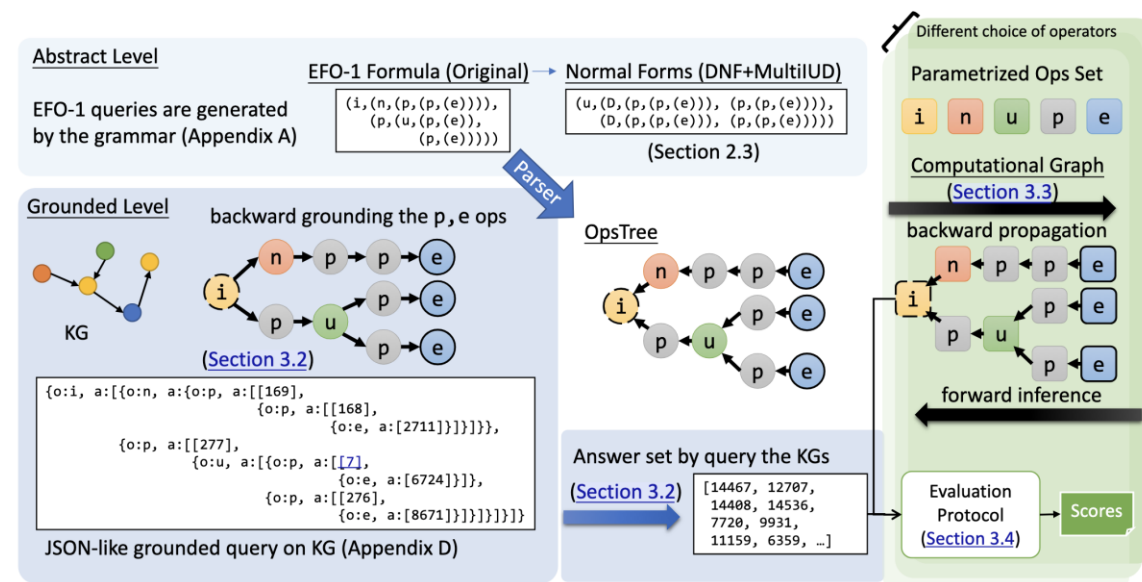
1. “geometric” intuition such as box in 1D
2. The element-wise comparison

So that it supports both good intuition and fuzzy logic t-norms

Summary of systems of neural set operations

Query representation

- A natural decomposition for combinatorial generalizability.
- The choices of operators can vary.
- Several equivalent forms exist.
- More details of such systems and how different choice affects the performance can be found in Wang et al., (NeurIPS'21)



Summary of systems of neural set operations

- Design of embedding spaces
 - Intuitions are essential for parameterization.
 - Many intuitions from KG embeddings also apply, pointing to many recent works.
- Training and inferences
 - Inference is efficient; it can be done with almost constant data complexity
 - Because neural operations are usually in fixed dimensions.
 - Nearest Neighbor lookup can be sped up.
 - Training is standard by negative samples.
 - Recent works about meta-learning have helped to improve generalizability.
 - Key takeaway: *Don't consider meta models; consider meta operators.* (Yin et al., 2024)

Theoretical concerns:

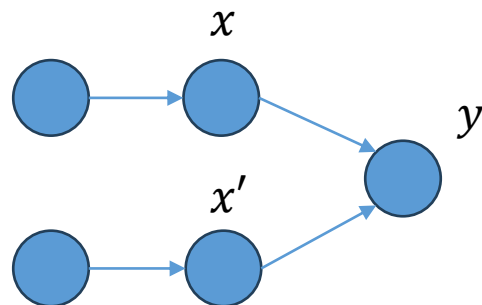
- What queries are we solving?
 - Strategy I (completion and search):
 - Existential First Order (EFO).
 - Strategy II (end-to-end learning, neural set operator):
 - “The queries that can be expressed in Trees”, let’s call it Tree-Form (TF) query.

Is TF the same as EFO ?

“The queries that can be expressed in Trees”

To formal define TFQ, we should describe logical queries that expresses

- Atomic query (h, r, ?)
- Set projection
- Set complement
- Set intersection
- Set union



Let $\mathcal{T}$ be the set of all TFQs, then

- $S_{atomic} \in \mathcal{T}$
- If $\phi(z) \in \mathcal{T}$, then
$$\exists z. \phi(z) \wedge r(z, y) \in \mathcal{T}$$
- If $\phi \in \mathcal{T}$, then
$$\neg\phi(y) \in \mathcal{T}$$
- If $\phi, \psi \in \mathcal{T}$, then
$$\phi(y) \wedge^* \psi(y) \in \mathcal{T}$$
$$\phi(y) \vee^* \psi(y) \in \mathcal{T}$$

Note *: the existential variables in $\phi(y)$ and $\psi(y)$ are assumed to be not shared.

TFQ vs EFO: The difference in definition

Tree-form query

Let $\mathcal{T}$ be the set of all TFQs, then

- $S_{atomic} = \{r(h, y) : r \in \mathcal{R}, h \in \mathcal{E}\} \in \mathcal{T}$
 h is an entity, y is a variable

- If $\phi \in \mathcal{T}$, then
 $\neg\phi(y) \in \mathcal{T}$

- If $\phi, \psi \in \mathcal{T}$, then
 $\phi(y) \wedge^* \psi(y) \in \mathcal{T}$
 $\phi(y) \vee^* \psi(y) \in \mathcal{T}$

Note*: the existential variables in $\phi(y)$ and $\psi(y)$ are assumed to be not shared.

- If $\phi(z) \in \mathcal{T}$, then
 $\exists z. \phi(z) \wedge r(z, y) \in \mathcal{T}$

Existential First Order (EFO) query

Let $\mathcal{Q}$ be the set of all EFO query, then

- $F_{atomic} = \{r(t_1, t_2), r \in \mathcal{R}\} \in \mathcal{Q}$
 t_1 and t_2 are either entities or variables.

- If $\phi \in F_{atomic}$, then
 $\neg\phi(y) \in \mathcal{Q}$

- If $\phi, \psi \in \mathcal{Q}$, then
 $\phi(y) \wedge \psi(y) \in \mathcal{Q}$
 $\phi(y) \vee \psi(y) \in \mathcal{Q}$

Note1: the existential variables in $\phi(y)$ and $\psi(y)$ can be shared.

- If $\phi \in \mathcal{Q}$ and x is a variable, then
 $\exists x. \phi \in \mathcal{Q}$

Note2: we can always use this rule to make sure there is only one free variable (non-quantified). This subset is also known as EFO₁.

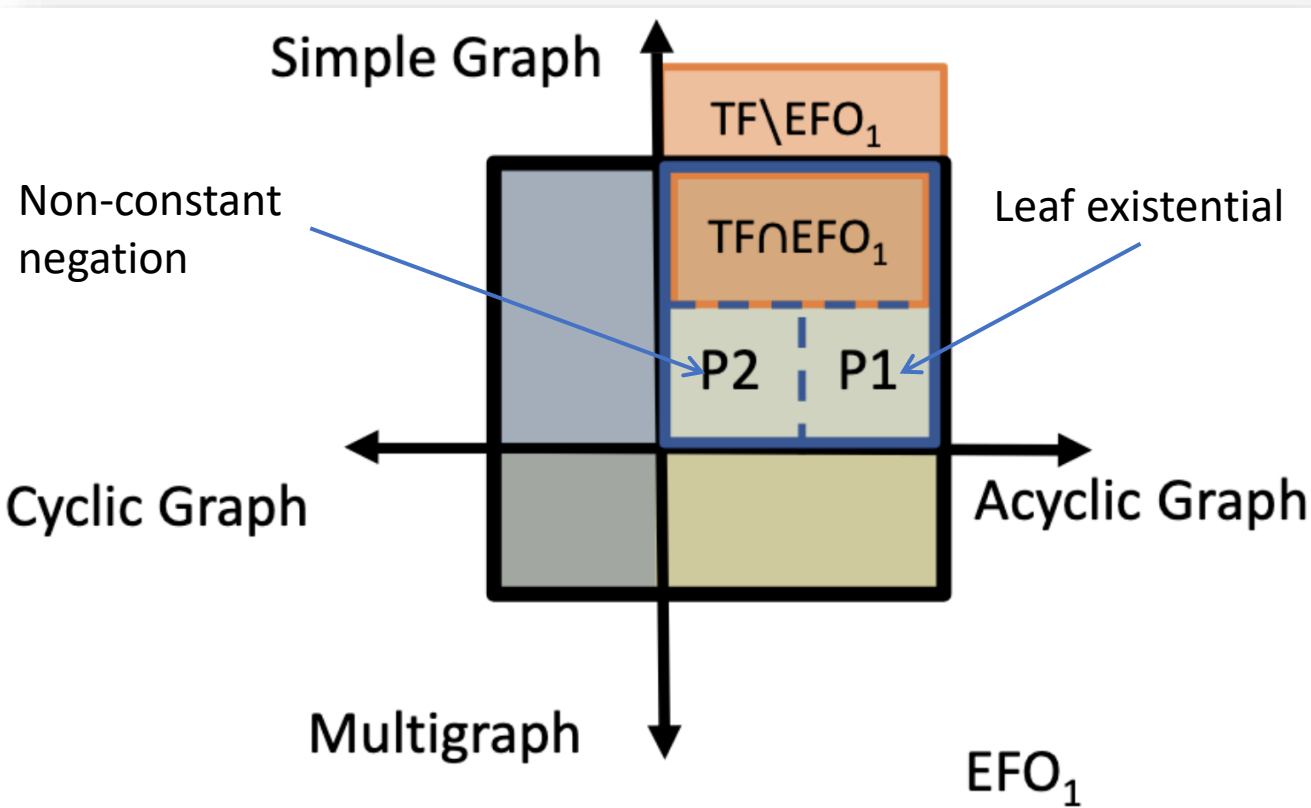
$$CQ_j(y; x_1, \dots, x_n) = y. \exists x_1, \dots, \exists x_n. \bigwedge_{k=1, \dots, M_j} a_{jk}$$

TFQ vs EFO: Are TFQ $\mathcal{T}$ the same as EFO $\mathcal{Q}$?

$\mathcal{T}$ –

Construction:

- $r_1(a, y) \in \mathcal{T}$
- $\exists x. r_1(a, x) \wedge r_2(x, y) \in \mathcal{T}$
- $\neg \exists x. r_1(a, x) \wedge r_2(x, y) \in \mathcal{T}$
- $\phi(y) = \forall x. \neg r_1(a, x) \wedge r_2(x, y) \in \mathcal{T}$
- There is a universal existential.

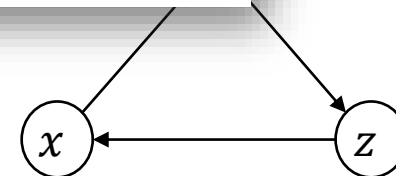


not empty

$(z, x) \in \mathcal{Q}$

$(y) \wedge r(y, z) \wedge r(z, x) \in \mathcal{Q}$

but not a tree.

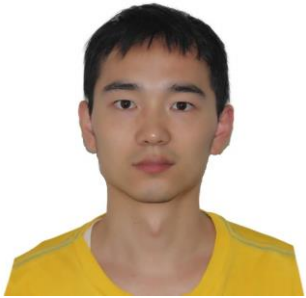


Neural reasoning for logical query: summary

- Logical query in DB:
 - Solves various forms of logical queries
- Logical Reasoning in Enriched KG
 - Strategy I: first complete, then conduct logical inference.
 - Strategy II: mixture of completion and logical inference.
- Still far away from SPARQL-compatible queries
 - Some key features are missing, such as:
 - multiple variables to answer. (Yin et al., 2023)
 - operators such as count, distinct, sort, ...

Coffee Break: Resume in 11 AM

New Frontiers of Knowledge Graph Reasoning: Recent Advances and Future Trends



Lihui Liu

lihuil2@illinois.edu



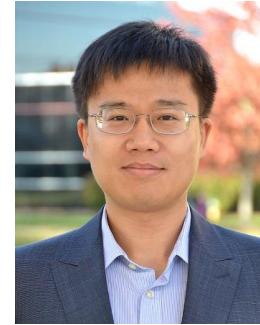
Zihao Wang

zwanggc@cse.ust.hk



Jiaxin Bai

jbai@connect.ust.hk



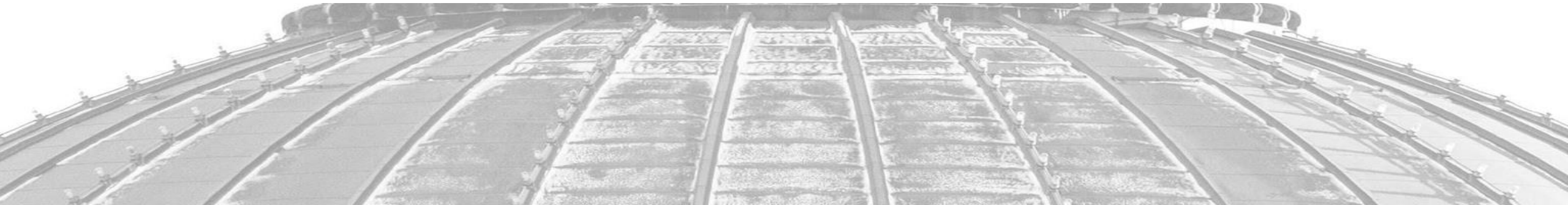
Yangqiu Song

yqsong@cse.ust.hk



Hanghang Tong

htong@illinois.edu



New Frontiers of Knowledge Graph Reasoning: Recent Advances and Future Trends

Part IV: Neural Reasoning Beyond Entities and Relations (1)

Yangqiu Song

Department of CSE, HKUST



KnowComp Group
Understanding the World by Computational Knowledge

Roadmap

Part I: Knowledge Graph Reasoning: Basic Concepts and Techniques

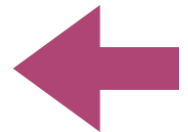
Part II: Recent Advance #1: Neural Reasoning for Natural Language Queries

Part III: Recent Advance #2: Neural Reasoning for Logical Queries

Part IV: Recent Advance #3: Neural Reasoning Beyond Entities and Relations

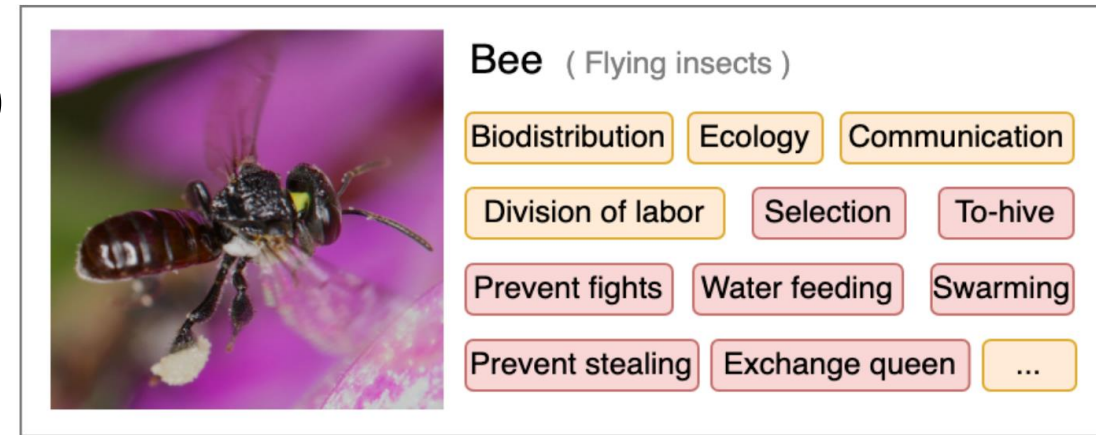
Part V: Recent Advance #4: LLM+KGR

Part VI: Open Challenges and Future Directions



Scope of Knowledge

- **Awareness of facts**
 - “Knowing that”: the truth of propositions
 - E.g., the population of Singapore is 5.92 million (2023)
- A possession of **practical skills**
 - “Knowing how”: understanding how to perform certain actions; procedural knowledge
 - E.g., How to hive bees?
- An experiential acquaintance: **familiarity with individuals and situations**
 - “Knowing by acquaintance”: directly perceiving an object, being familiar with it, or otherwise coming into contact with it.
 - E.g., by eating Durian, one becomes acquainted with the taste of it



The aspects in orange boxes are aspects that convey the knowledge of “know-what” or “know-why”, while those in red boxes convey the knowledge of “know-how”.

Source: Kuaipedia

Commonsense Knowledge in AI

- "Commonsense knowledge includes the basic facts about **events** (including actions) and **their effects**, ... **how it is obtained**, **facts about beliefs and desires**. It also includes the basic **facts about objects and properties**. " – John McCarthy
- "While to the average person the term 'commonsense' is regarded as synonymous with '**good judgement**', the AI community it is used in a technical sense to refer to the **millions of basic facts and understandings possessed by most people**." --ConceptNet
 - "Such knowledge is typically omitted from social communications", e.g.,
 - **If you forget someone's birthday, they may be unhappy with you.**
- Meanwhile, it is not invariably true
 - "a person is larger than a dog"



Source: Zorba - The World's Largest Dog Ever Lived

The Three-dimensional Development of Knowledge Graphs in Computer Science

The development of the Web

In the past years the development of knowledge graph has been largely changed (or revolutionized) by deep learning and NLP

Web 3.0

Web 2.0

Web 1.0

G#1: Entity-
Based KGs

G#2: Text-
Rich KGs

G#3: Dual
Neural KGs

Generations
of KGs

Xin Luna Dong. Generations of Knowledge Graphs: The crazy ideas and the business impact. *VLDB*, 2023. Invited paper for **VLDB Women in Database Research Award**. [[Paper](#)][[Talk](#)]

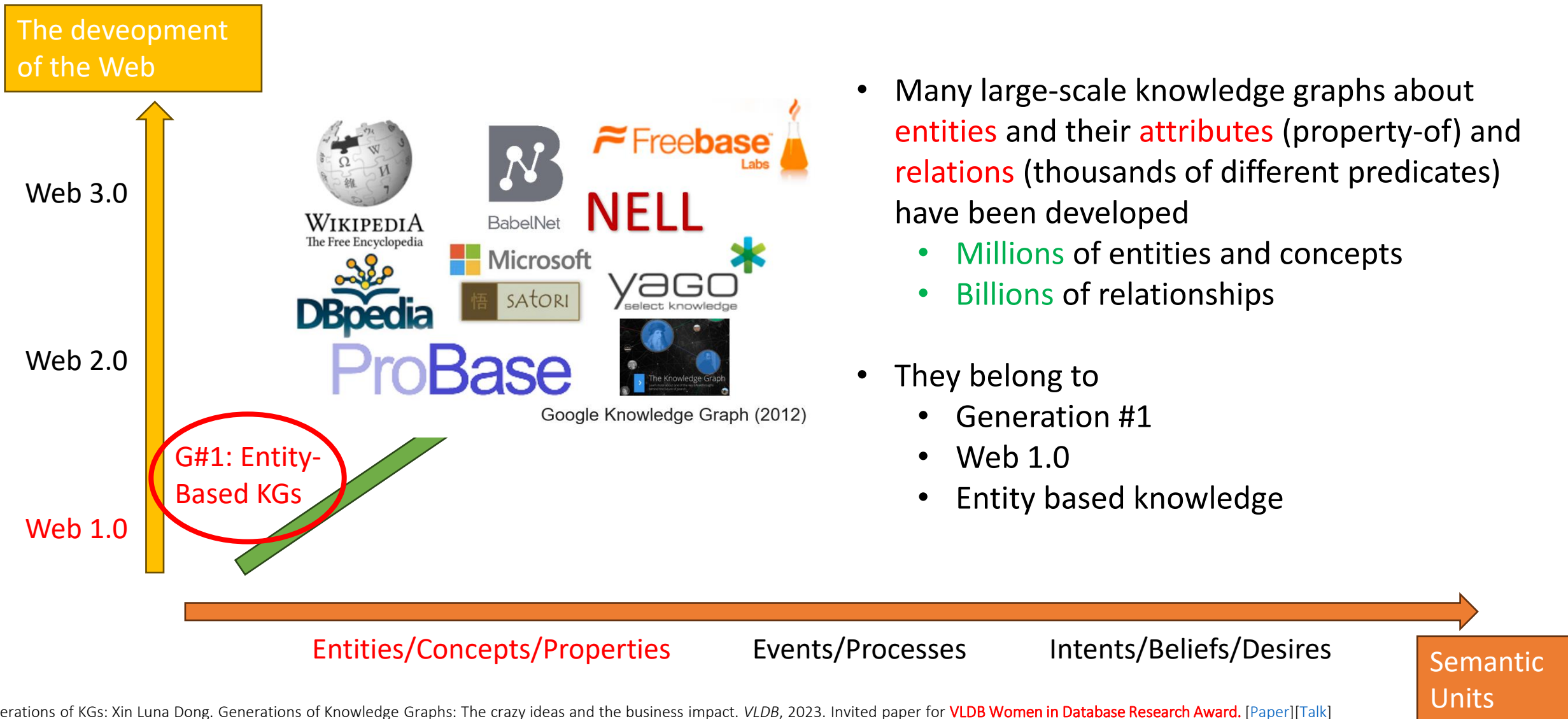
Entities/Concepts/Properties

Events/Processes

Intents/Beliefs/Desires

Semantic
Units

The Three-dimensional Development of Knowledge Graphs in Computer Science



The development of the Web

Web 3.0

Web 2.0

Web 1.0



Generic KG

Movie,
Music,
Book,
etc.

Product
Graph

(Hardline, softline,
consumables, etc.)

G#
Ne

G#2: Text-Rich KGs

G#1: Entity-Based KGs

Product Graph

- Generation #2
- Web 1.0
- Entity (product) based knowledge



ASER

ConceptNet

An open, multilingual knowledge graph

Concept Graph

Eventuality Graph

```

graph TD
    wake_up_morning((wake up in morning))
    alarm_clock((alarm clock))
    get_to_bed_early((get to bed early))
    in_house((in house))
    kitchen_table((kitchen table))
    chair((chair))
    eat_breakfast((eat breakfast))
    full_stomach((full stomach))
    read_newspaper((read newspaper))
    chew_food((chew food))
    drink_coffee((drink coffee))
    check_email((check e-mail))
    yawn((yawn))

    wake_up_morning -- UseOf --> alarm_clock
    wake_up_morning -- MotionOf --> get_to_bed_early
    in_house -- LocationOf --> kitchen_table
    kitchen_table -- PartOf --> chair
    wake_up_morning -- PrerequisiteOf --> eat_breakfast
    eat_breakfast -- EffectOf --> full_stomach
    eat_breakfast -- SubeventOf --> read_newspaper
    eat_breakfast -- SubeventOf --> chew_food
    eat_breakfast -- SubeventOf --> drink_coffee
    eat_breakfast -- SubeventOf --> check_email
    yawn -- SubeventOf --> wake_up_morning
  
```

- ASER/Knowlywood

- Generation #2
- Web 1.0
- Event based knowledge

- ConceptNet

- Generation #2
- Web 1.0
- Entity and event based knowledge

Entities/Concepts/Properties

Events/Processes

Intents/Beliefs/Desires

Semantic Units

The Three-dimensional Development of Knowledge Graphs in Computer Science

The development of the Web

Web 3.0

Web 2.0

Web 1.0

G#1: Entity-Based KGs

G#2: Text-Rich KGs

G#3: Dual Neural KGs

Generations of KGs



“Language models as knowledge bases”
Neural KGs/Memory Networks/NGDBs

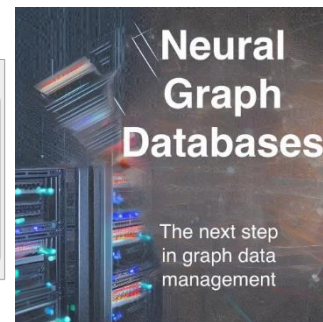
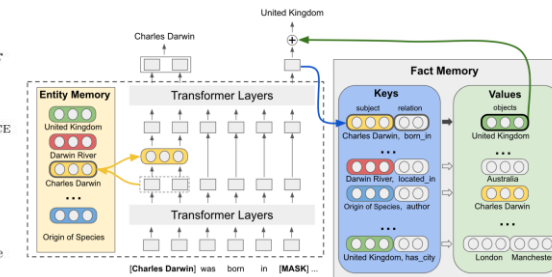
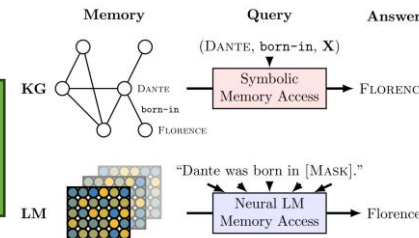


Figure of NGDBs

- Zihao Wang's part
 - Generation #3
 - Web 1.0
 - Entity (product) based knowledge



- Jiaxin Bai's part
 - Generation #3
 - Web 1.0
 - Event and Intention based knowledge

Entities/Concepts/Properties

Events/Processes

Intentions/Beliefs/Desires

Semantic Units

The Three-dimensional Development of Knowledge Graphs in Computer Science

The development of the Web

Web 3.0

Web 2.0

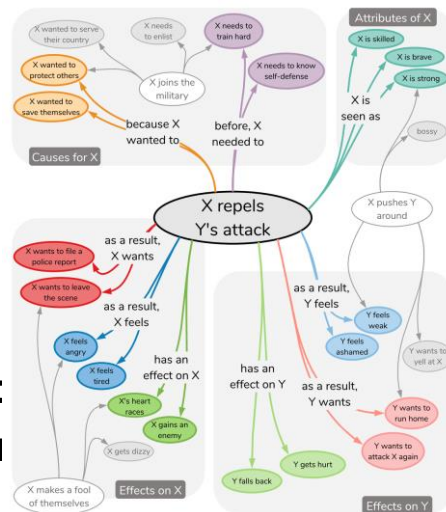
Web 1.0

G#1: Entity-
Based KGs

G#2: Text-
Rich KGs

G#3: Neu

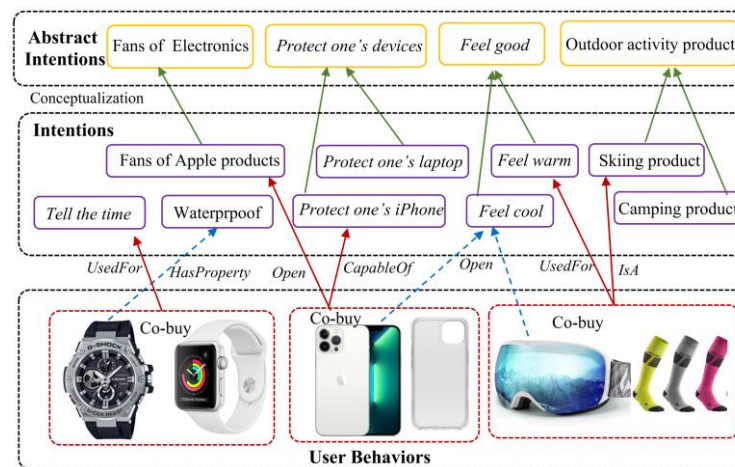
ATOMIC



ATOMIC

- Generation #2
- Web 1.0
- Intention and event (cause-effect) knowledge

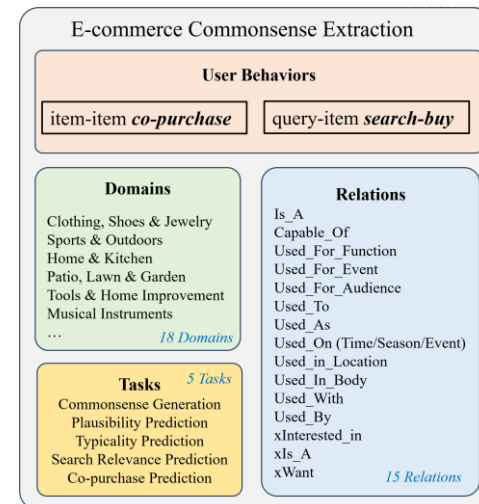
FolkScope



FolkScope

- Generation #2
- Web 2.0
- Intention knowledge

amazon COSMO



COSMO

- Generation #2
- Web 2.0
- Intention knowledge

Entities/Concepts/Properties

Events/Processes

Intentions/Beliefs/Desires

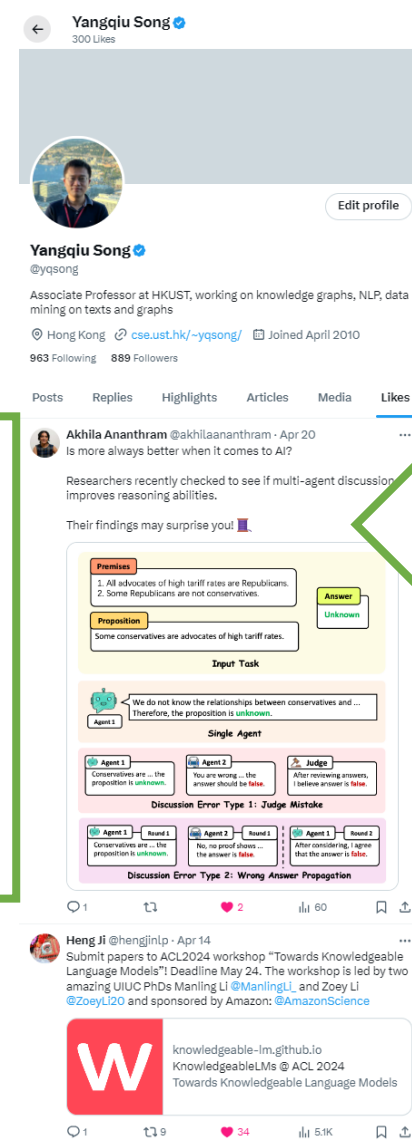
Semantic
Units

Why do we want to go beyond entities in Web2.0?

- User centric and user generated content (UGC)
 - Explicit incentives :
 - e.g., user reviews, posting blogs, tweets, photos, videos, etc.
 - Implicit incentives :
 - user behaviors, e.g., following, browsing, clicking, searching, upvoting, etc.



Why do I **post** this tweet?
Sell my own work, or promote my student?



Why do I **like** this post? I like the content or promote my own work?

Why do we want to go beyond entities in Web2.0?

“I post this tweet: Sell my own work or promote my student?” Inference involves:

- **1. Theory of Mind**

- i.e., the development of **knowledge** that **others have beliefs, desires, and intentions** that are different from one's own
- Possessing a functional theory of mind is crucial for success in everyday human social interactions

- What makes us take actions?

- **Beliefs** and **desires** are mediated by **intentions** which in turn controls human's **actions (or speech)** (Kashima et al., 1998)
- Intentions are implicit

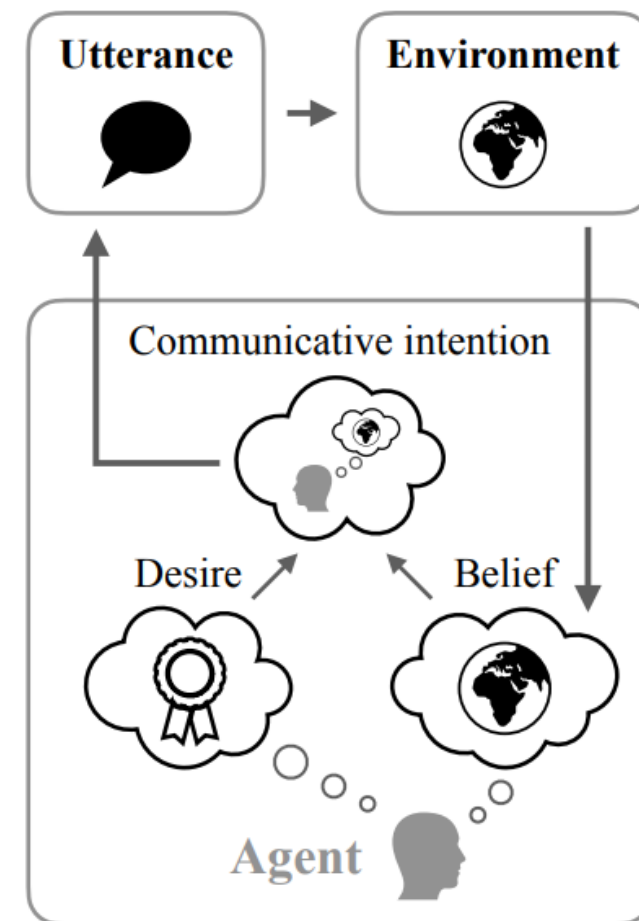


Figure taken from
Andreas (2022)

<https://en.wikipedia.org/wiki/Intention>

https://en.wikipedia.org/wiki/Theory_of_mind

Jacob Andreas: Language Models as Agent Models. EMNLP (Findings) 2022: 5769-5779

Kashima, Yoshihisa, Allison McKintyre, and Paul Clifford. "The Category of the Mind: Folk Psychology of Belief, Desire, and Intention." Asian Journal Of Social Psychology 1, no. 3 (December 1998): 289–313.

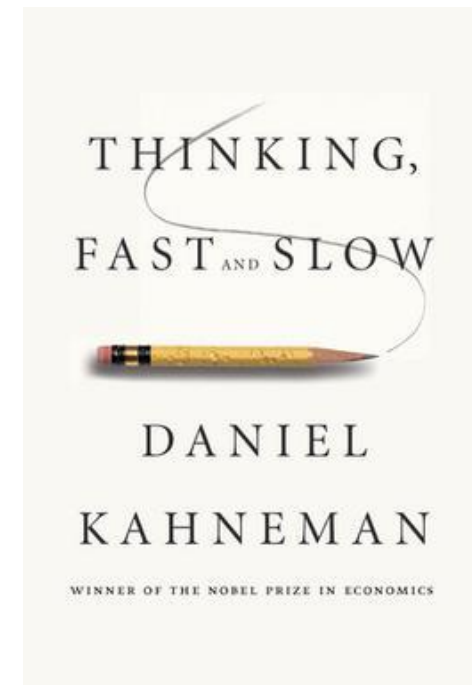
Why do we want to go beyond entities in Web2.0?

“I post this tweet: My student gets more visible or not?”

Inference involves:

- **2. System II Processing**

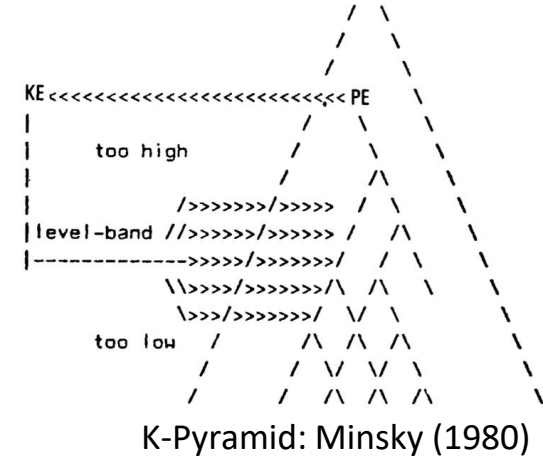
- We need to equip machine learning systems with “slow, logical, sequential, conscious, linguistic, algorithmic, planning, reasoning”
- Particularly, such a system requires the “understanding of how **actions** interact with **changes (of states) in distribution**”
 - “Agents face **non-stationarities**”
 - Conditioned on “different places, times, sensors, actuators, goals, policies, etc”



Why Graphs?

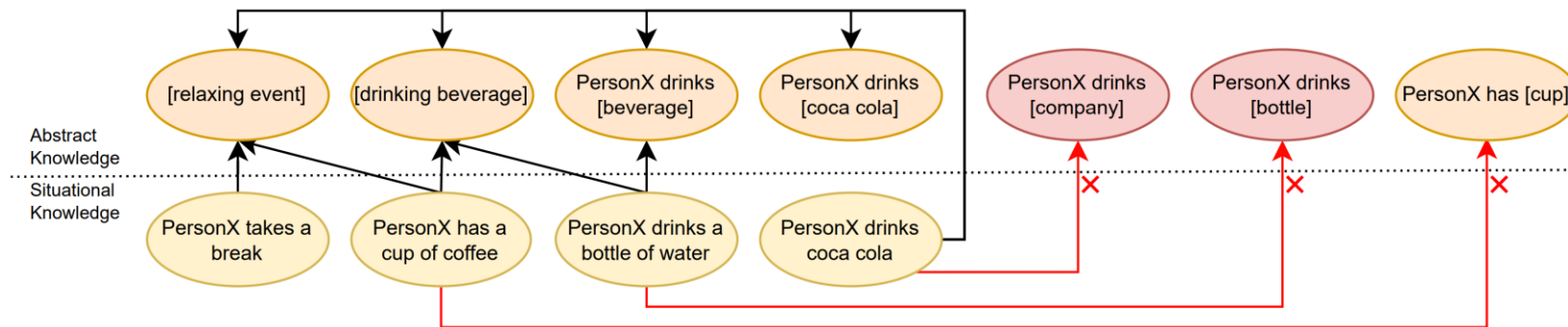
- The K-Line Theory (Minsky, 1980)

- More than **ontology**: categories include substances, properties, relations, states of affairs, and events
- **Mental states** in our memory are also in a hierarchical structure **beyond an ontology**; described as a **K-pyramid**



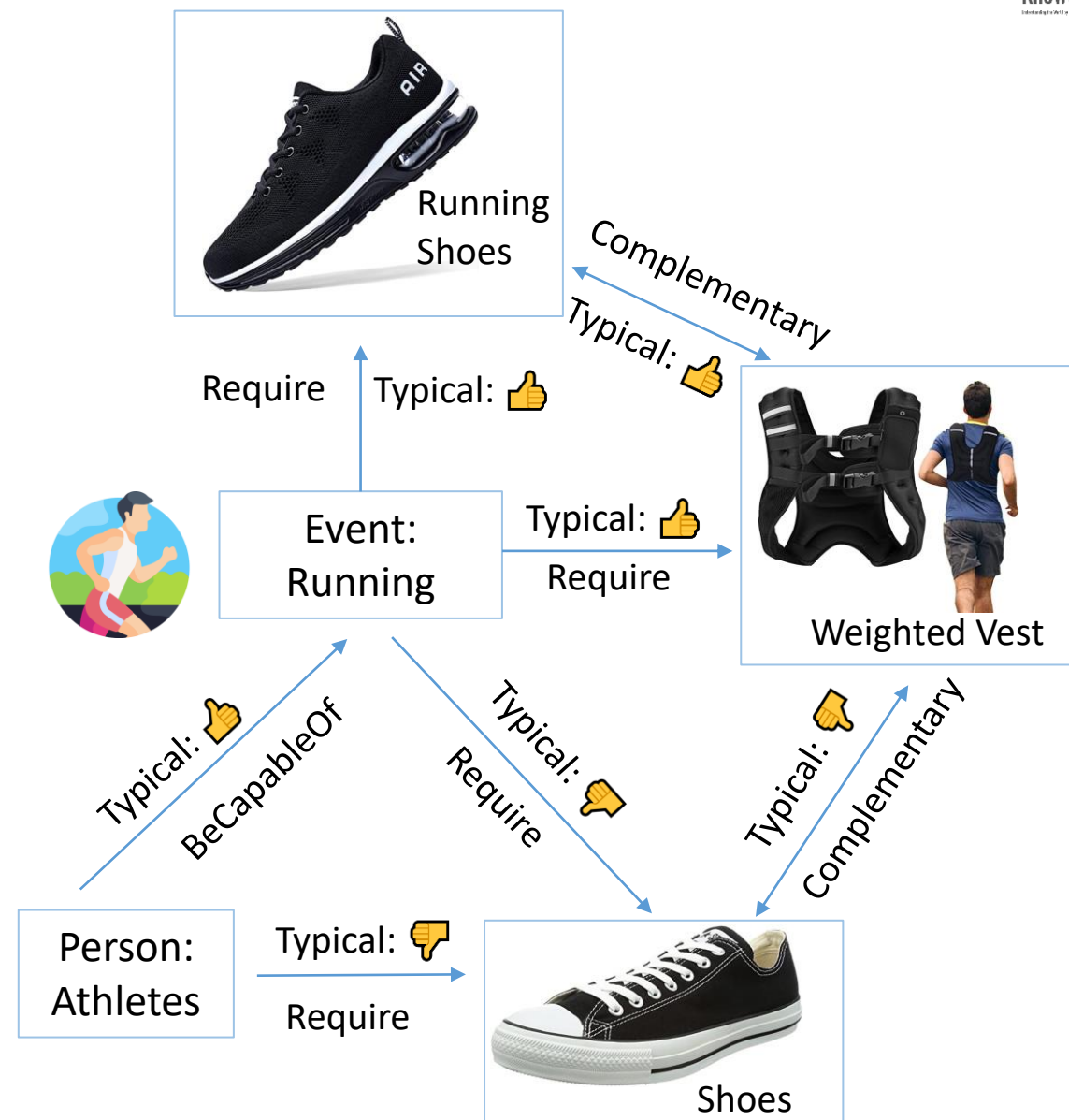
- We need the **right level** and **right perspective** of abstraction

- **Different levels of abstractness**: “PersonX drinks coca cola” → “[drinking beverage],” “[event]”
- **Different perspectives**: “Coca cola” → “[sugary beverage],” “[phosphate containing beverage],” “[iced drink],” not in a strict taxonomy
 - PersonX drinks [iced drink], xReact, refreshed
 - PersonX drinks [sugary beverage], xEffect, gain weight



Why Graphs?

- Sometimes we need concrete, symbolic, and globally referenced knowledge (Edge et al., 2024)
- Ability of commonsense reasoning with high complexity
 - NP-complete problems, e.g., Max-Sat (Chalier et al., 2022), subgraph matching or counting, subset sum, etc.
- The trade-offs between scalability and computational complexity
 - Intentions can be pre-stored and indexed to be more efficiently accessed online



FolkScope and COSMO: User-centric Intention KGs

- AI generated knowledge graph construction framework

Actions on E-commerce Platform

Explainable recommendation

UserU bought [A] because

Query-item relevance

PersonX searches [Q] to

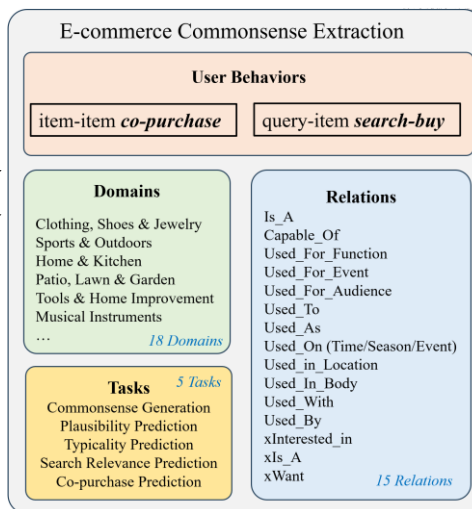
User behavior rationalization

PersonX bought [A] and [B] because

PersonX searched [A] and bought [B] because

Prompts

AI Generator



Human
Feedback

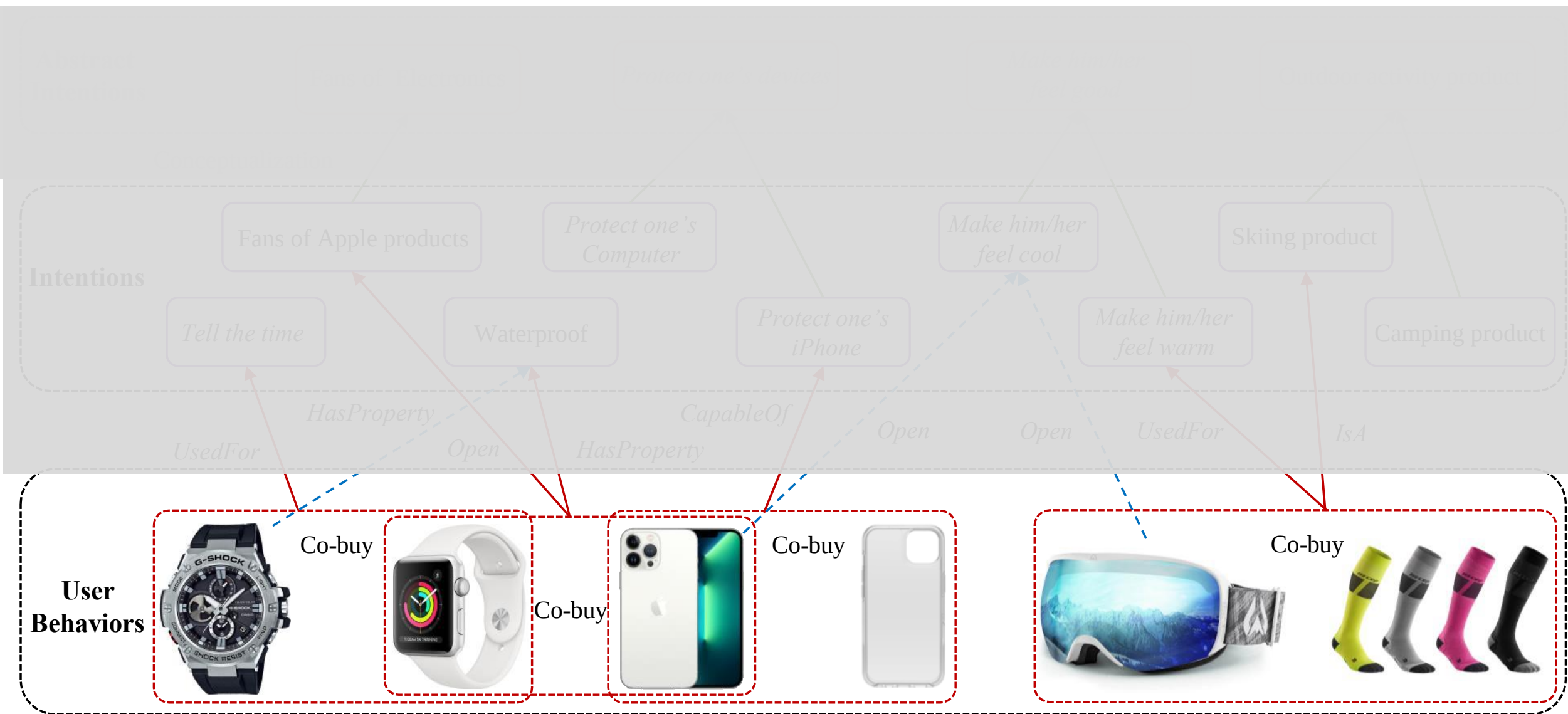
AIG-KG



Intention KG

Pattern mining, Filtering,
Normalization, Conceptualization

FolkScope: Collective Intention KG for Co-Purchases



Deployments: COSMO, Intention KG for Search-buy

- Efficient **feature store** and asynchronous **cache store**
- Effectively meets Amazon's **restricted search latency** requirements while maintaining **storage costs** comparable to real-time serving for the **majority of traffic**

Search Query Navigation

- **A/B tests** carried out over **several months** in total
- approximately **10% of Amazon's U.S. traffic**
- a notable **0.7% relative increase** in product sales
- translating to **hundreds of millions of dollars** in annual revenue surge.

Search query

The image shows a screenshot of an Amazon search results page for the query 'baking'. A red box highlights the search bar containing 'baking'. Below the search bar, there are filter buttons for 'Food Mixer', 'Baking Mix', 'Baking Paper', and 'Baking Pan'. The 'RESULTS' section shows several sponsored products. A red dashed line connects the search bar to a red box labeled 'Concept' which contains the text 'Baking tools' over a collage of baking-related images. To the right, three product listings are shown, each enclosed in a red box. The first listing is for a 'USA Pan Bakeware Rectangular Cake Pan, 9 x 13 inch, Nonstick & Quick Relea...' with a price of \$21.99. The second listing is for a 'Nonstick Bakeware Set w/ Baking Pans, Baking Sheets, Cookie Sheets, Muffin Pan, ...' with a price of \$79.99. The third listing is for a 'Nutrichef w/Heat Red Silicone Handles, Oven Safe, 3 Piece Set' with a price of \$17.93. A red arrow labeled 'Click' points from the 'Baking tools' concept box to the third product listing.

Search query

prime

Filters

Food Mixer Baking Mix Baking Paper Baking Pan

RESULTS

Sponsored

50 PCS Air Fryer Disposable Paper Liner, 6.3 Inches Air Fryer Liners, Non-stick Roun...

50PCS

★ ★ ★ ★ ★ 1

\$7.99

Save 10% with coupon

prime Get it as soon as Thu, Jun 2

FREE Shipping over \$25 by Amazon

Exclusive to Amazon

Concept

Baking tools

Click

prime

Filters

Sponsored

USA Pan Bakeware Rectangular Cake Pan, 9 x 13 inch, Nonstick & Quick Relea...

★ ★ ★ ★ ★ 9,579

\$21.99

prime Get it as soon as Today, May 31

FREE Shipping over \$25 by Amazon

Item

Sponsored

Nonstick Bakeware Set w/ Baking Pans, Baking Sheets, Cookie Sheets, Muffin Pan, ...

★ ★ ★ ★ ★ 863

\$79.99 \$84.99

prime Get it as soon as Thu, Jun 2

FREE Shipping by Amazon

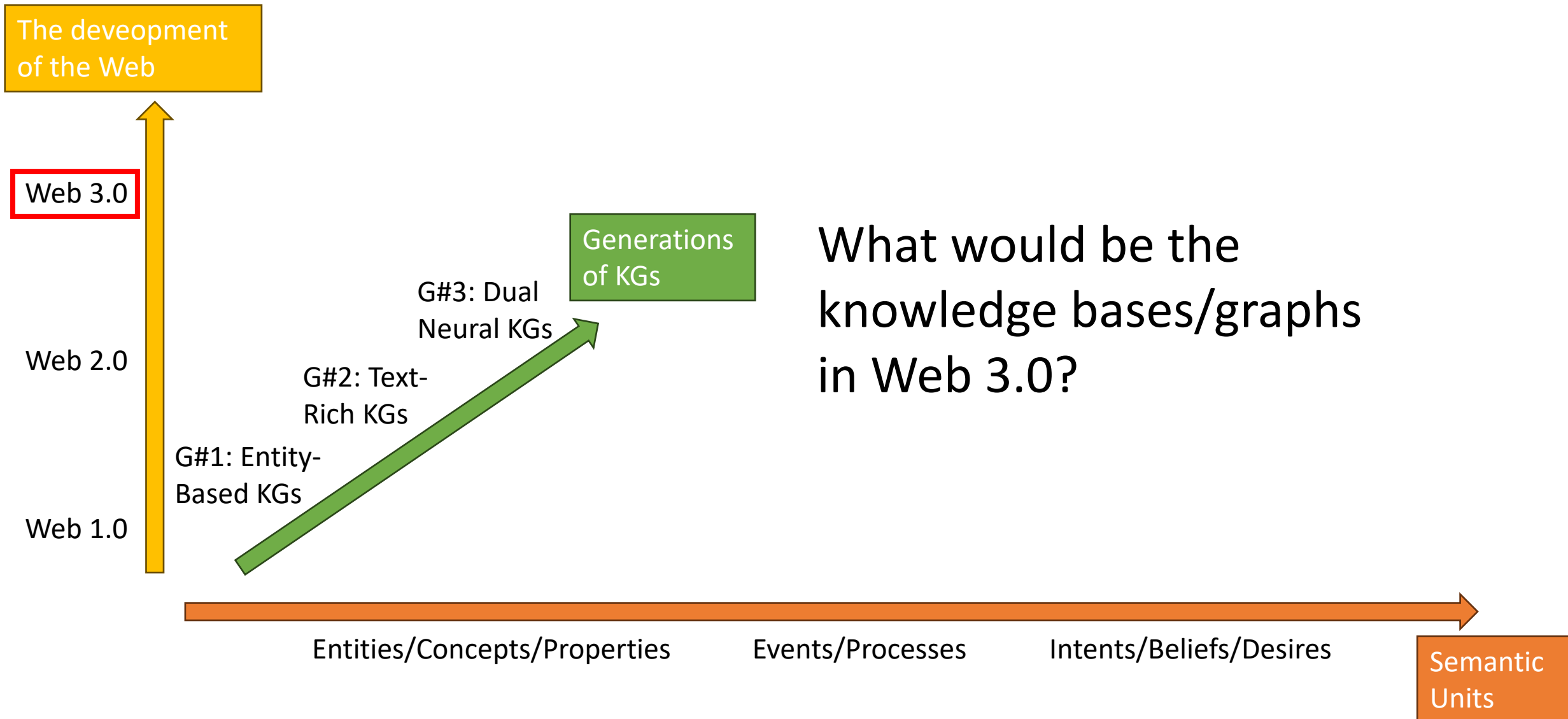
Best Seller

Nutrichef w/Heat Red Silicone Handles, Oven Safe, 3 Piece Set

★ ★ ★ ★ ★ 10,005

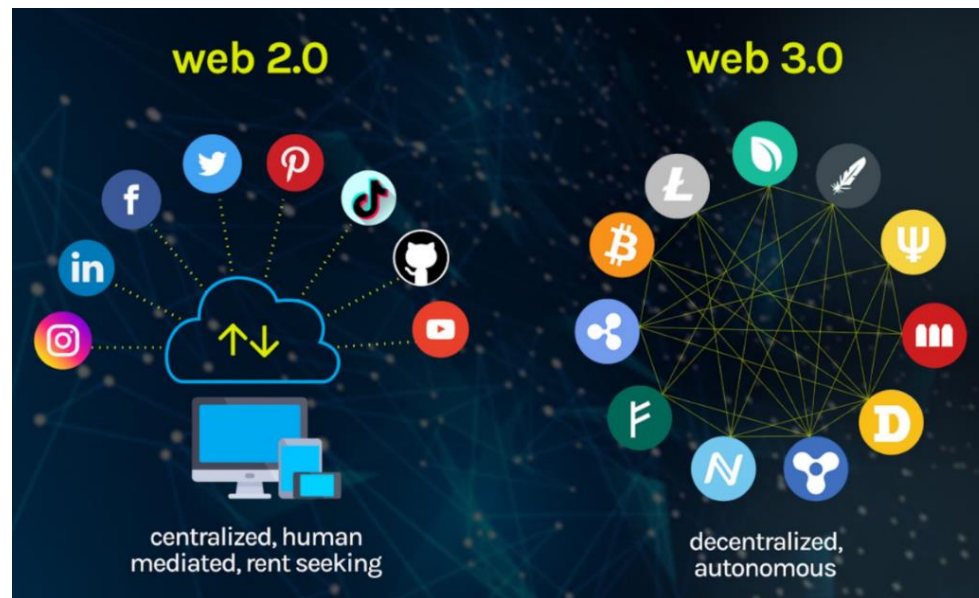
\$17.93 \$22.99

The Three-dimensional Development of Knowledge Graphs in Computer Science



An Outlook

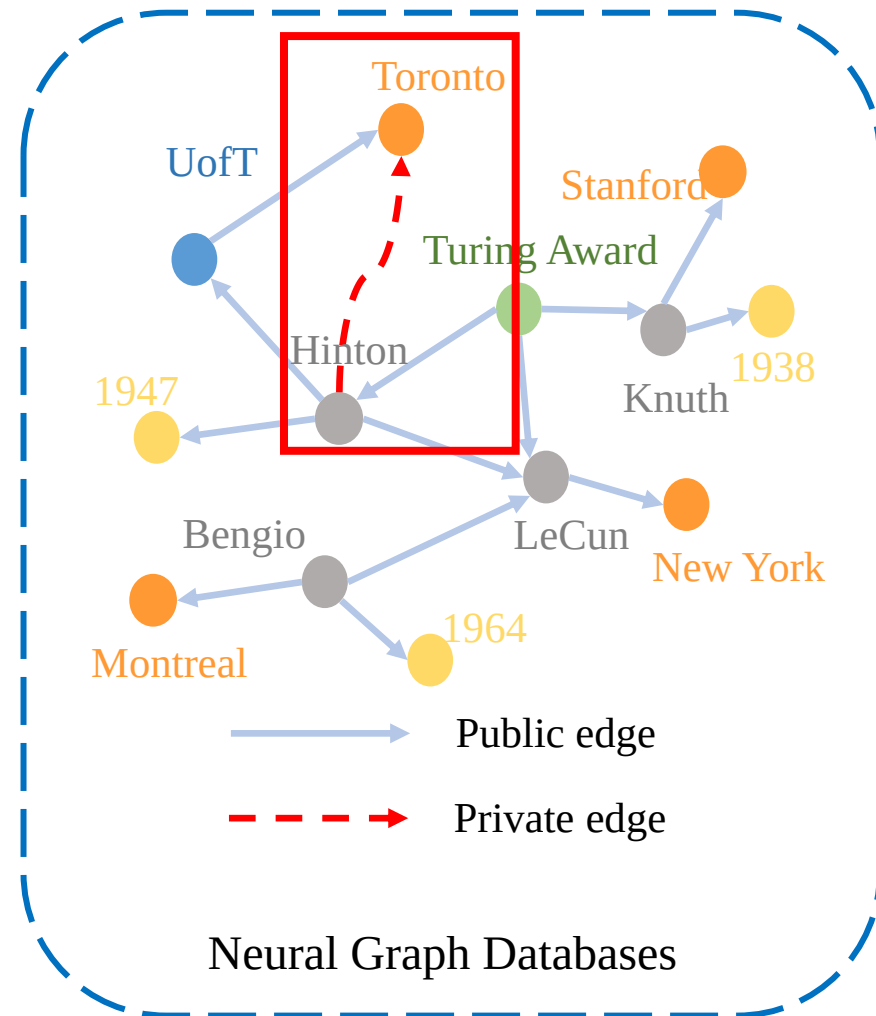
- From Web2.0 to Web3.0
 - Decentralized data: users own their (neural) knowledge bases/graphs
 - Monetize by users' data and time
 - Permissionless, trustless, but accessible to users' owned knowledge or data



- **Security and privacy** of data and knowledge is the key!

Privacy-preserved NGDBs

An attacker attempts to infer private information about Hinton's living place in the NGDBs. Attackers can leverage well-designed queries to retrieve desired privacy. The intersection of these queries can make a fair guess.

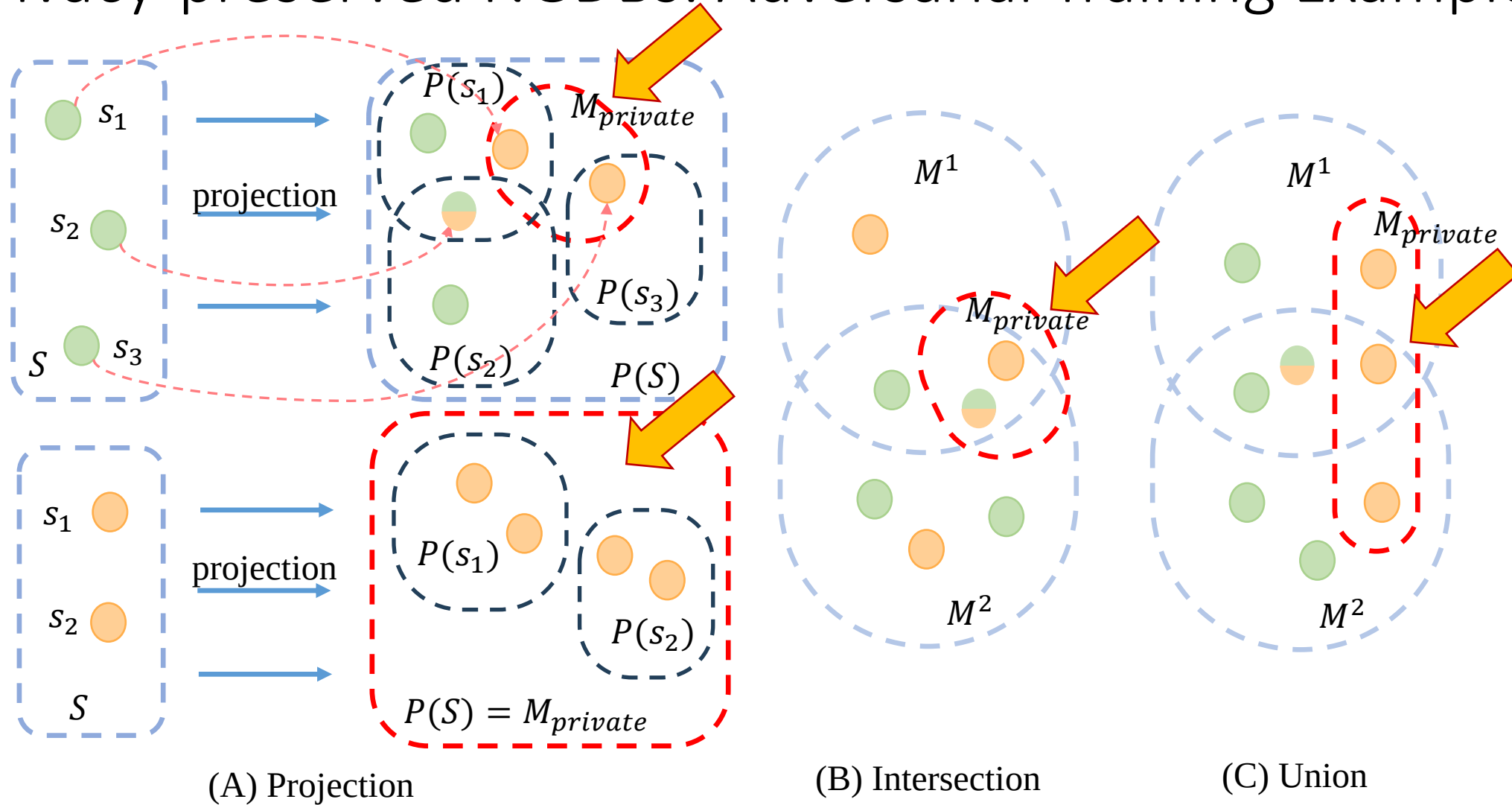


Query	$q = V_?. \exists V: \text{Win}(V, \text{Turing Award}) \wedge \text{BornIn}(V, 1938) \wedge \text{LiveIn}(V, V_?)$
Interpretation	Find where the Turing Award winner who was born in 1938 lived.

Complex Queries

Query	Answer
$q_1 = V_?. \text{LiveIn}(\text{Hinton}, V_?)$	Privacy risk query detection
$q_2 = V_?. \exists X_1, X_2: \text{Win}(X_1, \text{Turing Award}) \wedge \text{GreaterThanOr}(X_2, 1940) \wedge \text{BornIn}(X_1, X_2) \wedge \text{LiveIn}(X_1, V_?)$	Montreal, Toronto ...
$q_3 = V_?. \exists X_1: \text{CollabWith}(\text{LeCun}, X_1) \wedge \text{LiveIn}(X_1, V_?)$	Montreal, Toronto ...
$q_4 = V_?. \exists X_1, X_2: \text{Win}(X_1, \text{Turing Award}) \wedge \text{SmallerThan}(X_2, 1950) \wedge \text{BornIn}(X_1, X_2) \wedge \text{LiveIn}(X_1, V_?)$	Toronto , Stanford...

Privacy-preserved NGDBs: Adversarial Training Examples



(A) Projection

(B) Intersection

(C) Union

Green nodes denote non-private answers, orange nodes denote privacy-threatening answers, and orange-green nodes denote different privacy risks in subsets. Red dashed arrows denote privacy projection. The answers circled in red dashed line are at risk to leak privacy.

Conclusions for Part 4-1

- We have reviewed the frontier of recent development of knowledge graphs in terms of three dimensions
 - Three generations: entity, text, and neural
 - Semantic units: entity/attributes, events/processes, intent/desire/belief
 - **Web 1.0/2.0/3.0**
- More methods will be introduced in Part 4-2 by Jiaxin Bai
 - In terms of different types of semantic units



New Frontiers of Knowledge Graph Reasoning: Recent Advances and Future Trends

Part IV. Neural Reasoning Beyond Entities and Relations (2)

Jiaxin Bai

Department of CSE, HKUST



KnowComp Group
Understanding the World by Computational Knowledge

Roadmap

Part I: Knowledge Graph Reasoning: Basic Concepts and Techniques

Part II: Recent Advance #1: Neural Reasoning for Natural Language Queries

Part III: Recent Advance #2: Neural Reasoning for Logical Queries

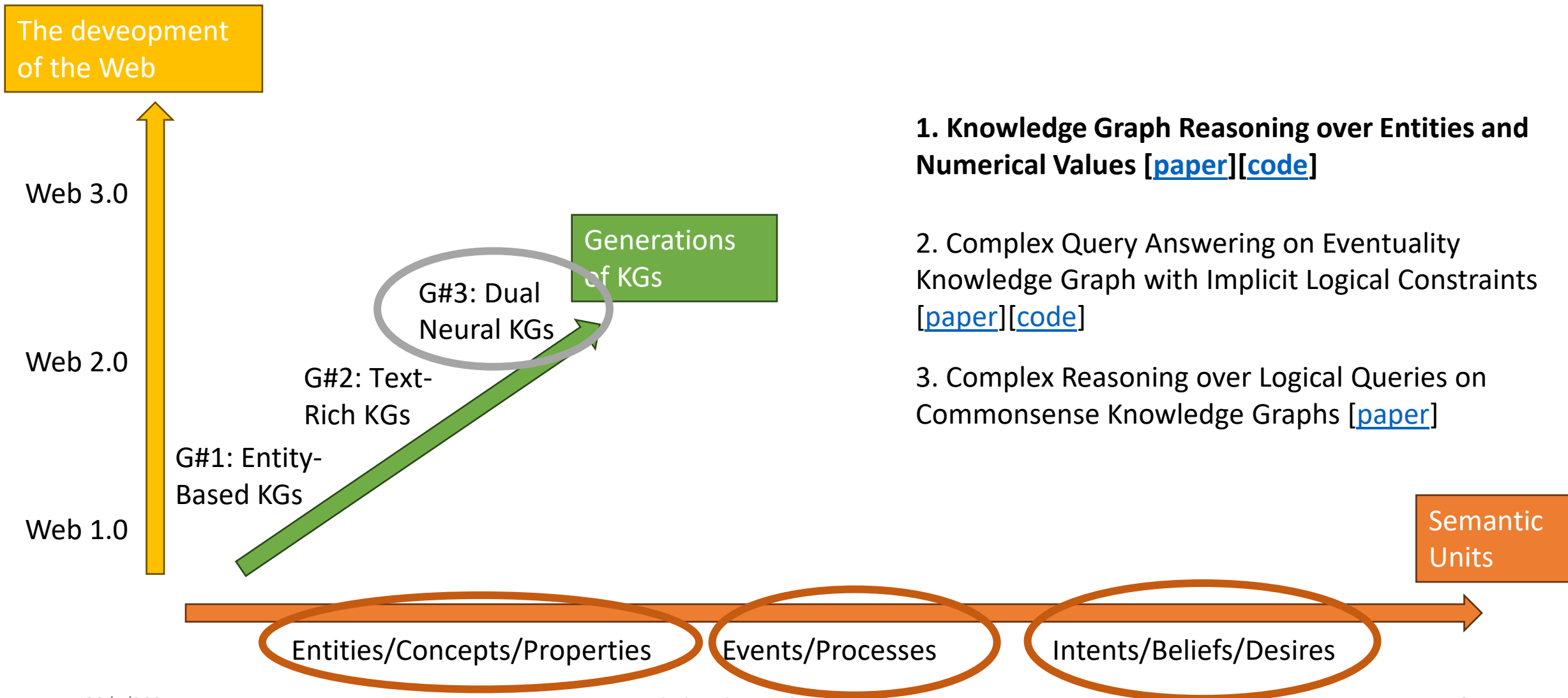
Part IV: Recent Advance #3: Neural Reasoning Beyond Entities and Relations

Part V: Recent Advance #4: LLM+KGR

Part VI: Open Challenges and Future Directions



The Three-dimensional Development of Knowledge Graphs in Computer Science

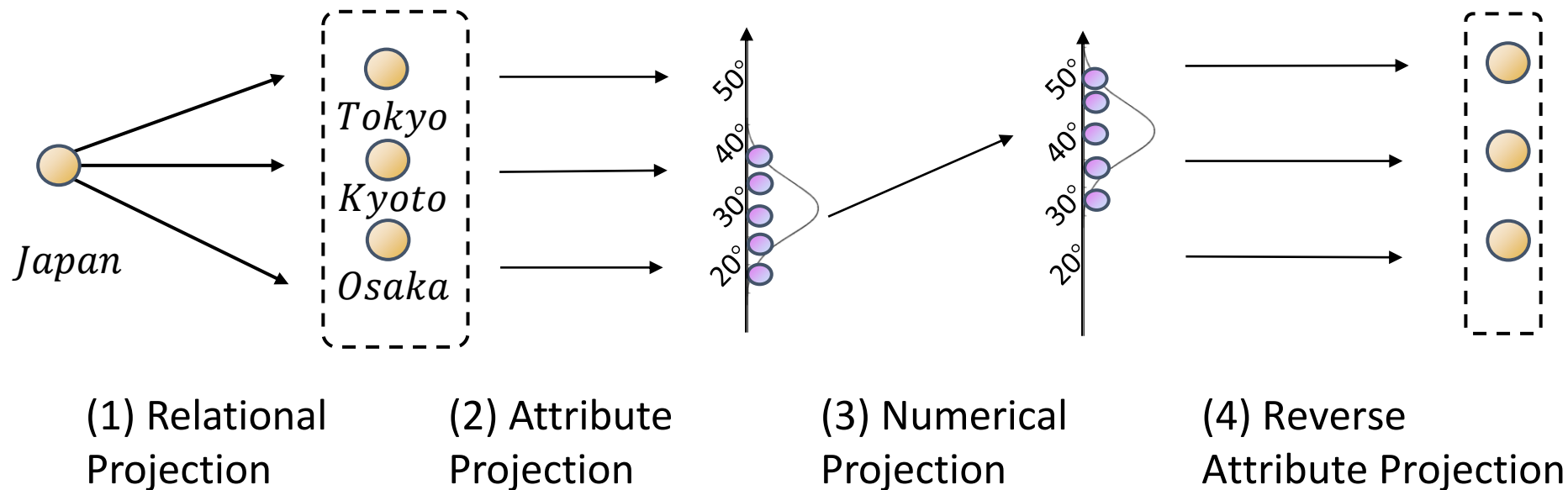
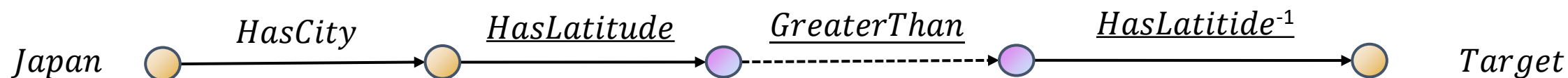


Numerical Complex Query Answering

Category	Complex Queries	Interpretations
Numerical CQA	$q_2 = V_? . \exists X_1, X_2: \text{Win}(V_?, \text{TuringAward})$ $\wedge \underline{\text{GreaterThan}}(1927, X_2) \wedge \underline{\text{BornIn}}(V_?, X_2)$	Find the Turing award winners that <u>is born before</u> the year of 1927.
Numerical CQA	$q_3 = V_? . \exists X_1, X_2: \text{LocatedIn}(V_?, \text{UnitedStates})$ $\wedge \underline{\text{HasLatitude}}(V_?, X_1)$ $\wedge \underline{\text{GreaterThan}}(X_1, X_2)$ $\wedge \underline{\text{HasLatitude}}(\text{Beijing}, X_2)$	Find the states in US that have <u>a higher latitudes</u> than Beijing.
Numerical CQA	q_4 $= V_? . \exists X_1, X_2, X_3: \text{LocatedIn}(V_?, \text{UnitedStates})$ $\wedge \underline{\text{HasPopulation}}(V_?, X_1)$ $\wedge \underline{\text{SmallerThan}}(X_1, X_2) \wedge \underline{\text{TimesByTwo}}(X_2, X_3)$ $\wedge \underline{\text{HasPopulation}}(\text{California}, X_3)$	Find the states in US that have a <u>twice smaller population</u> than California?

Number Reasoning Network

Find the cities that have a higher latitudes than Japanese cities.

$$q = V_? . \exists V_1, X_1, X_2: \underline{HasLatitude}(V_?, X_2) \wedge \underline{GreaterThan}(X_2, X_1) \wedge \underline{HasLatitude}(V_1, X_1) \wedge \underline{LocatedIn}(V_1, Japan)$$


Number Reasoning Network

(1) Relational Projection (rp):

Query Embedding $\rightarrow$ Entity Set

(2) Attribute Projection (ap):

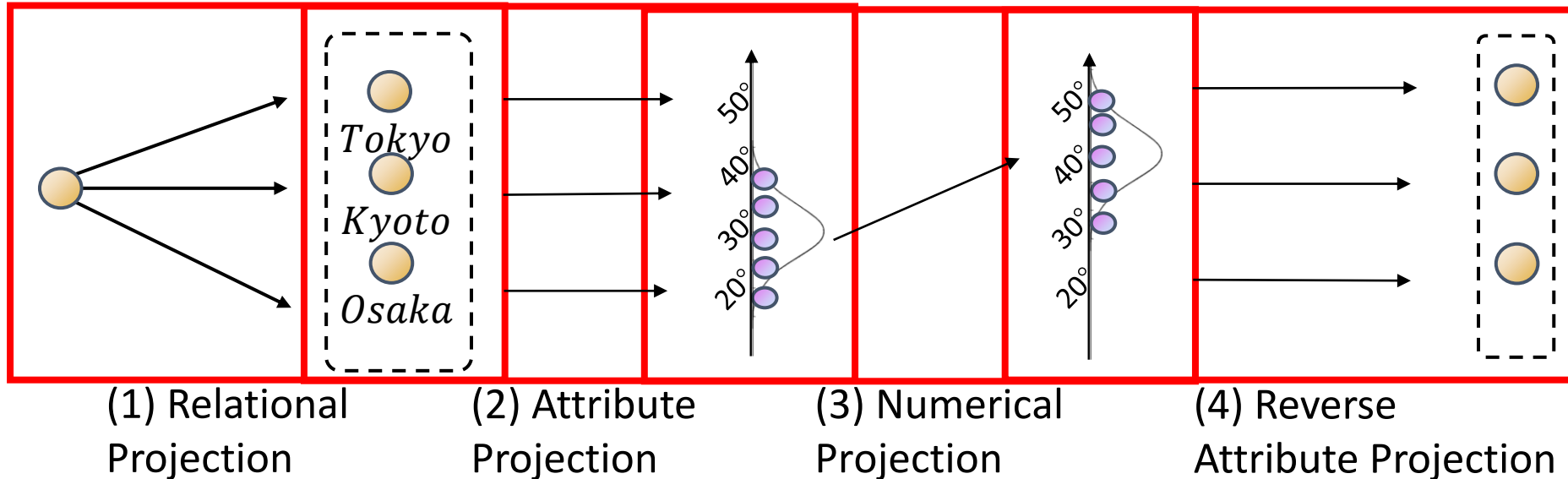
Query Embedding $\rightarrow$ Value Distribution

(3) Numerical Projection (np):

Value Distribution $\rightarrow$ Value Distribution

(4) Reverse Attribute Projection (rap):

Value Distribution $\rightarrow$ Query Embedding



Number Reasoning Network

(1) Relational Projection:
Adopted from the backbones:
GQE, Query2Box, Query2Particles.

(2) (3) (4) Other Projections: Gated Transitions

$$p_i = W_p^p q^i + b_p^p$$

Linear projection

$$z_i = \sigma(W_z^p e_a + U_z^p p_i + b_z^p)$$

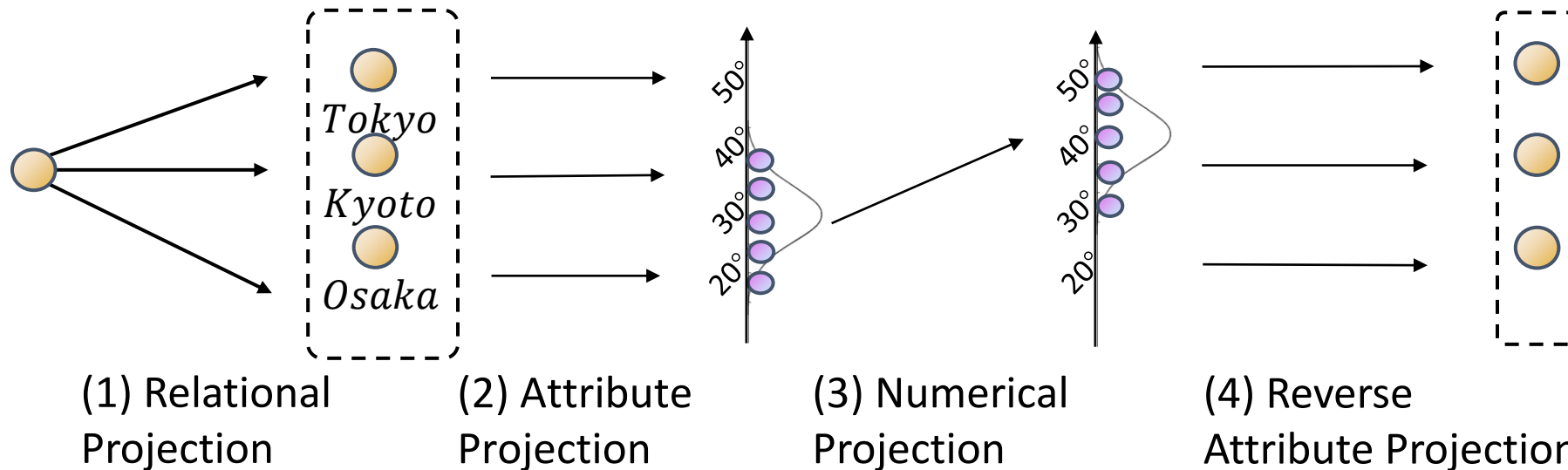
$$r_i = \sigma(W_r^p e_a + U_r^p p_i + b_r^p)$$

$$t_i = \varphi(W_h^p e_a + U_h^p (r_i \odot p_i) + b_h^p)$$

MLP

$$\theta_{i+1} = (1 - z_i) \odot p_i + z_i \odot t_i$$

Gate selection



Number Reasoning Network

Entity embeddings:

Adopted from the backbones:

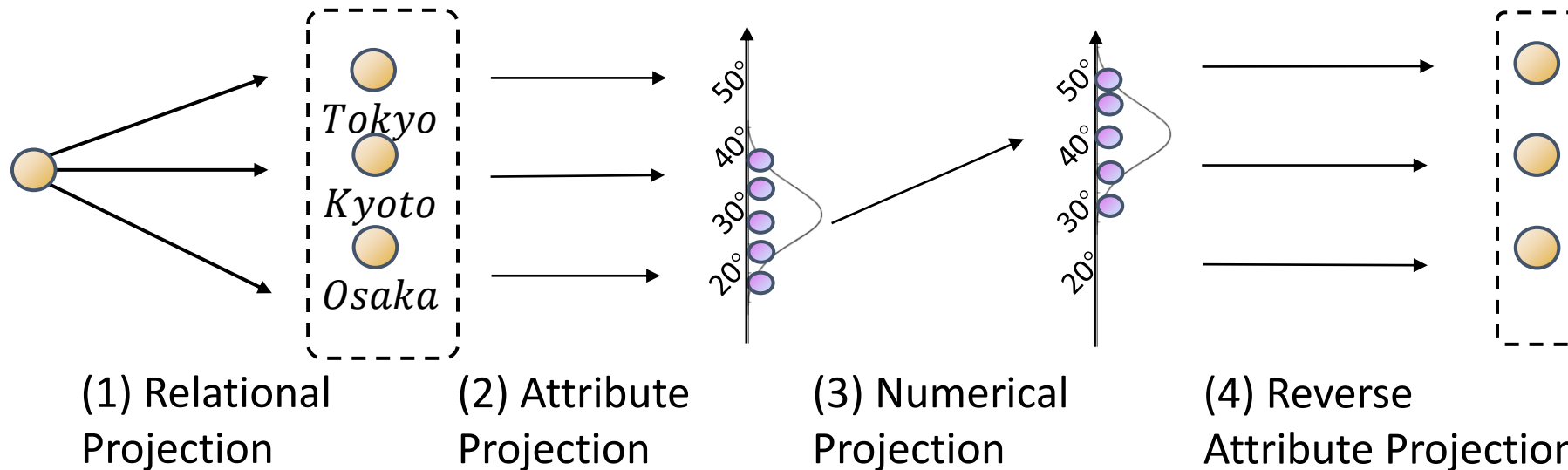
GQE, Query2Box, Query2Particles.

Input number embeddings

- DICE

- Sinusoidal

$$\psi(v)_d = \begin{cases} \sin^{d-1}(\alpha) \cos(\alpha) \\ \sin^D(\alpha) \end{cases} \quad \psi(v)_d = \begin{cases} \sin \frac{v}{v^{d/D}}, & d \equiv 0 \pmod{2} \\ \cos \frac{v}{v^{(d-1)/D}}, & d \equiv 1 \pmod{2} \end{cases}$$



Number Embeddings and Learning Objective

End-to-end training by Joint optimization of two losses:

$$L_A = \frac{1}{M} \sum_{j=1}^M (-\log p_{\theta_I^{(j)}}(\psi(v^{(j)})) - \log \phi_{t^{(j)}}(\theta_I^{(j)}))$$

The likelihood of the value

$v^{(j)}$ sampled from distribution of $\theta_I^{(j)}$

The likelihood of the distribution

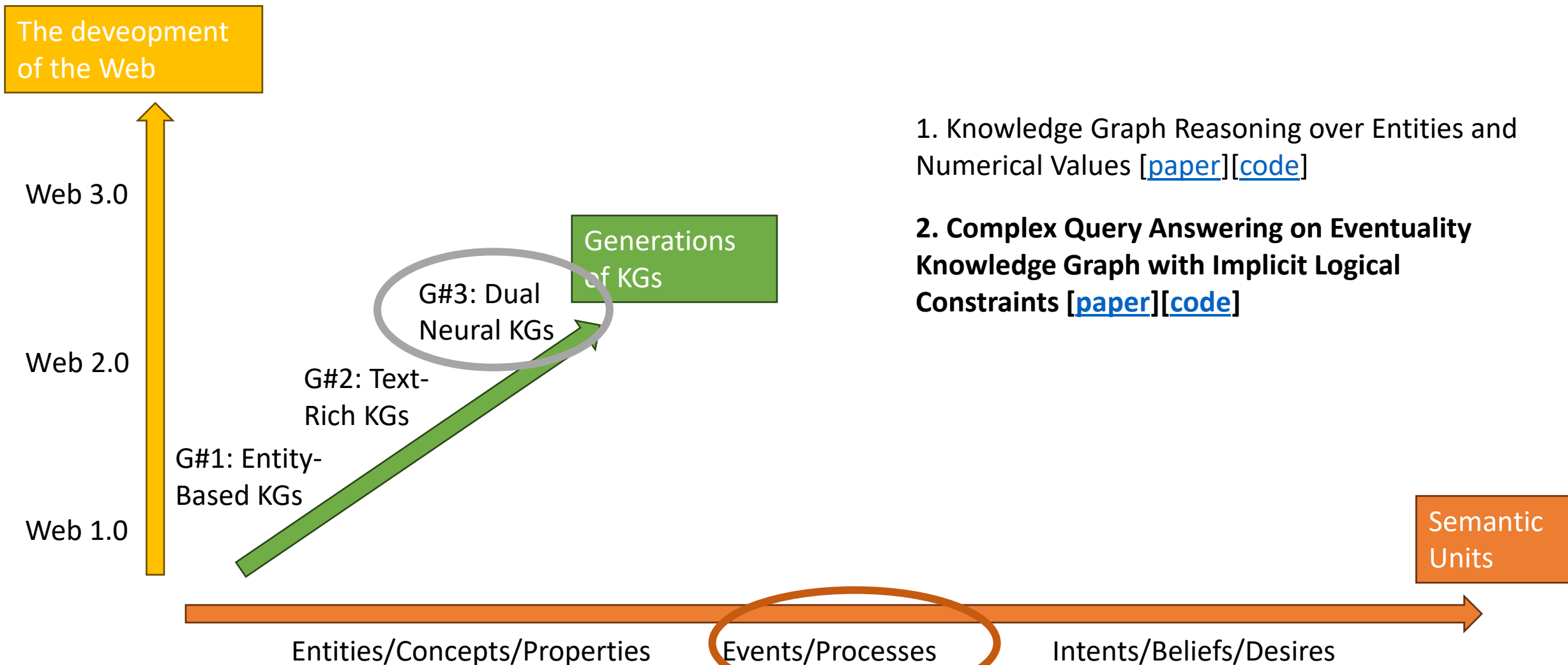
parameter $\theta_I^{(j)}$ is of type $t^{(j)}$

$$L_E = -\frac{1}{N} \sum_{j=1}^N \log p(q_I^{(j)}, v^{(j)})$$

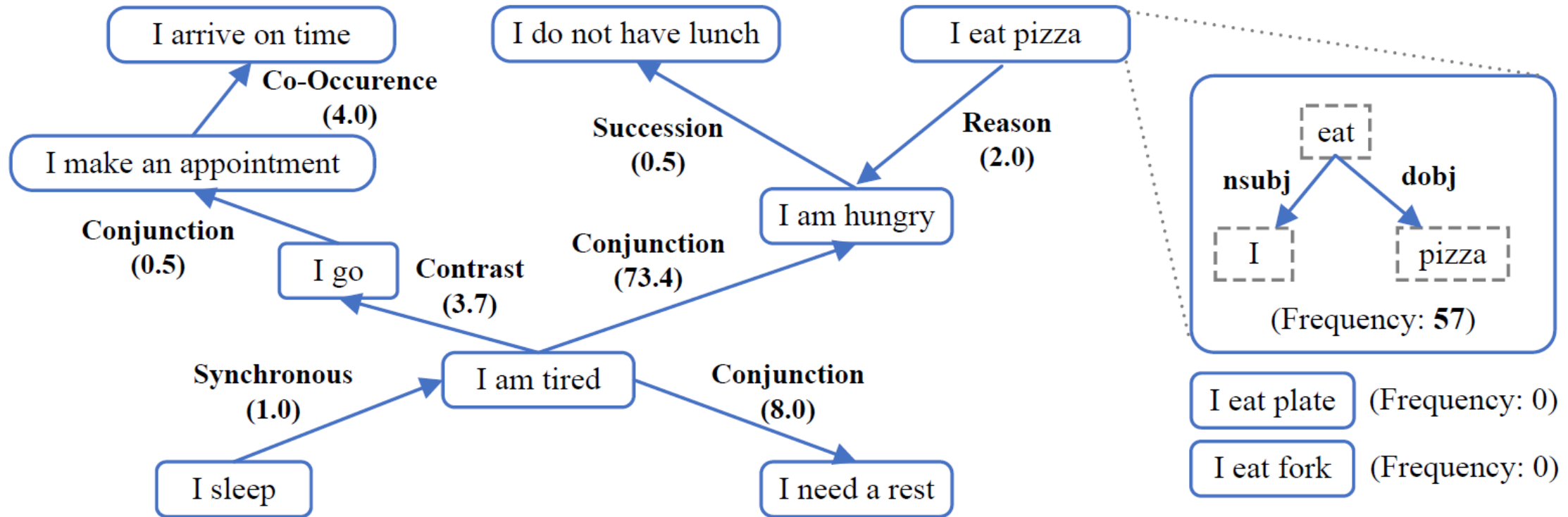
The likelihood of the entity $v^{(j)}$ is the answer of the query encoding $q_I^{(j)}$.

j is means the j -th sample, and I means the last step of distribution parameter encoding.

The Three-dimensional Development of Knowledge Graphs in Computer Science



ASER (Activities, States, Events, and their Relations)



Principle 1: Comparing semantic meanings by fixing grammar (Katz and Fodor, 1963)

Principle 2: The need of language inference based on ‘**partial information**’ (Wilks, 1975)

CQA on Eventuality Knowledge Graph

Complex query on eventuality graphs are **different** from the entity-relation graph

Whether and **when** the eventualities occur are important

Queries	Type	Interpretations
$q_1 = V_? . \exists V: \text{Interact}(V_?, V)$ $\wedge \text{Assoc}(V, \text{Alzheimer}) \wedge \text{Assoc}(V, \text{MadCow})$	Entity	Find the substances that interact with the proteins associated with Alzheimer's and Mad cow disease.
$q_2 = V_? . \text{Precedence}(\text{Food is bad}, \text{PersonX add soy sauce})$ $\wedge \text{Reason}(\text{Food is bad}, V_?)$	Eventuality	Food is bad before PersonX add soy sauce. What is the reason for food being bad?
$q_3 = V_? . \text{Precedence}(V_?, \text{PersonX go home})$ $\wedge \text{ChosenAlternative}(\text{PersonX go home}, \text{PersonX buy an umbrella})$	Eventuality	Instead of buying an umbrella, PersonX go home. What happened before PersonX go home?

Discourse Relations and Implicit Constraints

- PersonX did not eat anything because PersonX was full

Reason(PersonX did not eat anything, PersonX was full)

Occurrence
Constraint

$\eta(\text{PersonX did not eat anything}) \wedge \eta(\text{PersonX was full})$
 $\wedge \eta(\text{PersonX did not eat anything}) \leftarrow \eta(\text{PersonX was full})$

Temporal
Constraints

$\tau(\text{PersonX did not eat anything}) > \tau(\text{PersonX was full})$

$\eta(A) = 1$ if and only if it occurs
 $\tau(A) > \tau(B) : A$ happens after B

Discourse Relations and Implicit Constraints

- Food is bad before PersonX add soy sauce

Precedence(Food is bad, PersonX adds soy sauce)

Occurrence
Constraint

$\eta(\text{Food is bad}) \wedge \eta(\text{PersonX adds soy sauce})$

Temporal
Constraints

$\tau(\text{Food is bad}) < \tau(\text{PersonX adds soy sauce})$

$\tau(A) < \tau(B)$: A happens before B
 $\eta(A) = 1$ if and only if it occurs

Discourse Relations and Implicit Constraints

- Instead of buying an umbrella, PersonX go home

ChosenAlternative(buying an umbrella, *PersonX* go home)

Occurrence
Constraint

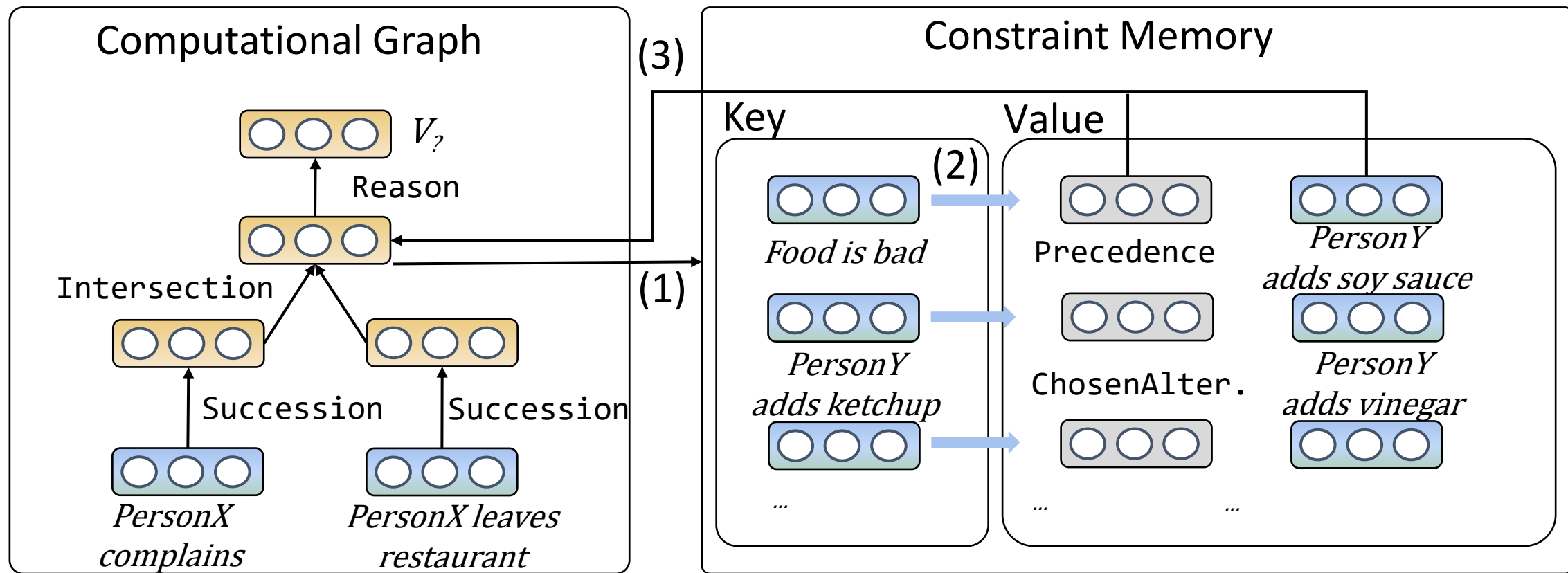
$\eta(\textit{PersonX go home}) \wedge \neg \eta(\textit{PersonX buy an umbrella})$

$\eta(A) = 1$ if and only if it occurs

Logical Constraints behind Discourse Relations

Discourse Relations	Semantics	Implicit Constraints	
		Occurrence Constraints	Temporal Constraints
Precedence(A, B)	A occurs before B.	$\eta(A) \wedge \eta(B)$	$\tau(A) < \tau(B)$
Succession(A, B)	A occurs after B happens.	$\eta(A) \wedge \eta(B)$	$\tau(A) > \tau(B)$
Synchronous(A, B)	A occurs at the same time as B.	$\eta(A) \wedge \eta(B)$	$\tau(A) = \tau(B)$
Reason(A, B)	A occurs because B.	$\eta(A) \wedge \eta(B) \wedge (\eta(A) \leftarrow \eta(B))$	$\tau(A) > \tau(B)$
Result(A, B)	A occurs as a result B.	$\eta(A) \wedge \eta(B) \wedge (\eta(A) \rightarrow \eta(B))$	$\tau(A) < \tau(B)$
Condition(A, B)	If B occurs, A.	$\eta(A) \rightarrow \eta(B)$	$\tau(A) > \tau(B)$
Concession(A, B)	B occurs, although A.	$\eta(A) \wedge \eta(B)$	-
Contrast(A, B)	B occurs, but A.	$\eta(A) \wedge \eta(B)$	-
Conjunction(A, B)	A and B both occur.	$\eta(A) \wedge \eta(B)$	-
Instantiation(A, B)	B is a more detailed description of A.	$\eta(A) \wedge \eta(B)$	-
Restatement(A, B)	A restates the semantics of B.	$\eta(A) \leftrightarrow \eta(B)$	-
Alternative(A, B)	A and B are alternative situations.	$\eta(A) \vee \eta(B)$	-
ChosenAlternative(A, B)	Instead of B occurs, A.	$\eta(A) \wedge \neg \eta(B)$	-
Exception(A, B)	A, except B.	$\neg \eta(A) \wedge \eta(B) \wedge (\neg \eta(B) \rightarrow \eta(A))$	-

Query Encoding with Constraint Memory



$$(1) s_{i,m} = \langle q_i, c_h^{(m)} \rangle$$

Computes the relevance of query embedding to the head of the memory key at position m .

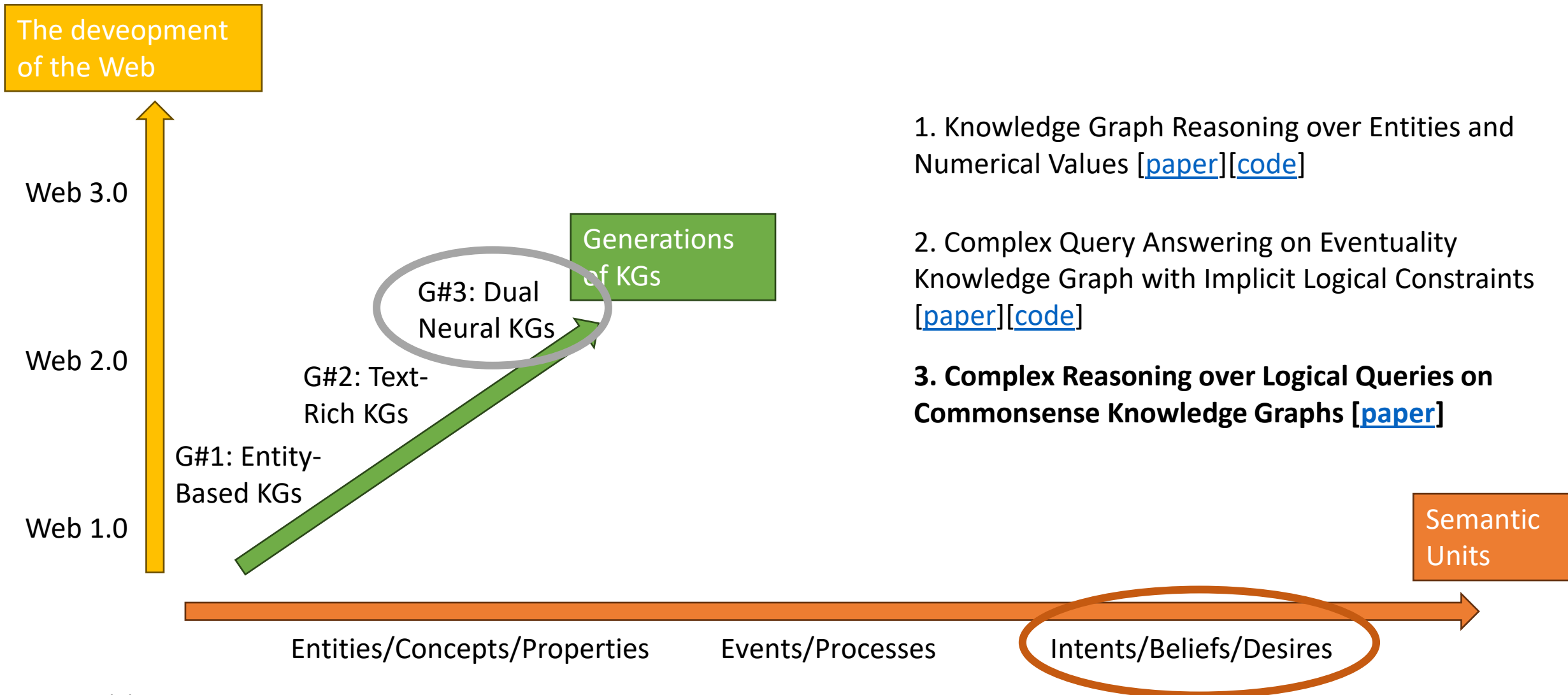
$$(2) v_i = \sum_{m=1}^M s_{i,m} (c_r^{(m)} + c_t^{(m)})$$

Computes the aggregated memory values across M memory cells with the importance weighted by relevance scores.

$$(3) q_i = q_i + \text{MLP}(v_i)$$

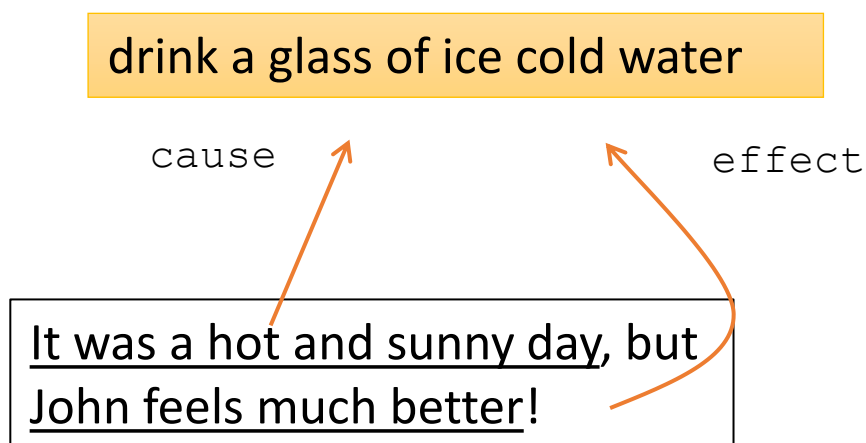
Computes the query embedding with memory values with the help of a MLP layer.

The Three-dimensional Development of Knowledge Graphs in Computer Science

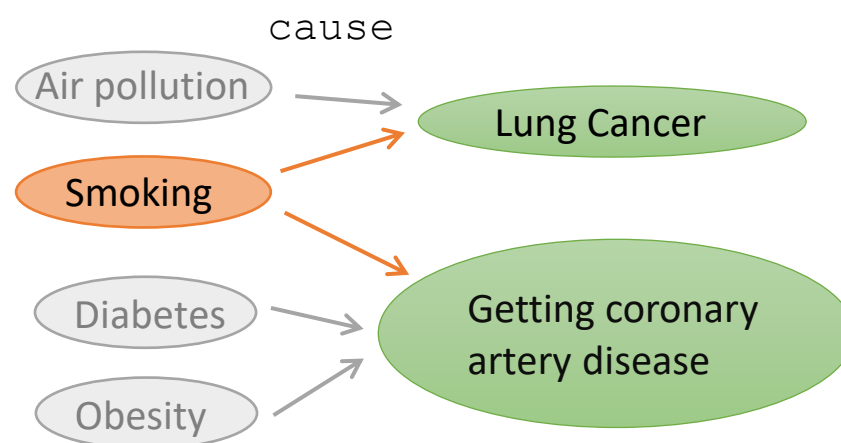


Moving towards intention based commonsense complex question answering

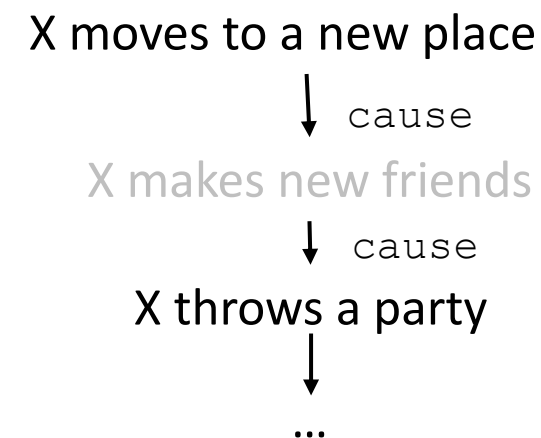
- Reasoning on real-world text and narratives requires complex reasoning over **multiple events**, and **inferring implicit context**.



Abduction



Common cause



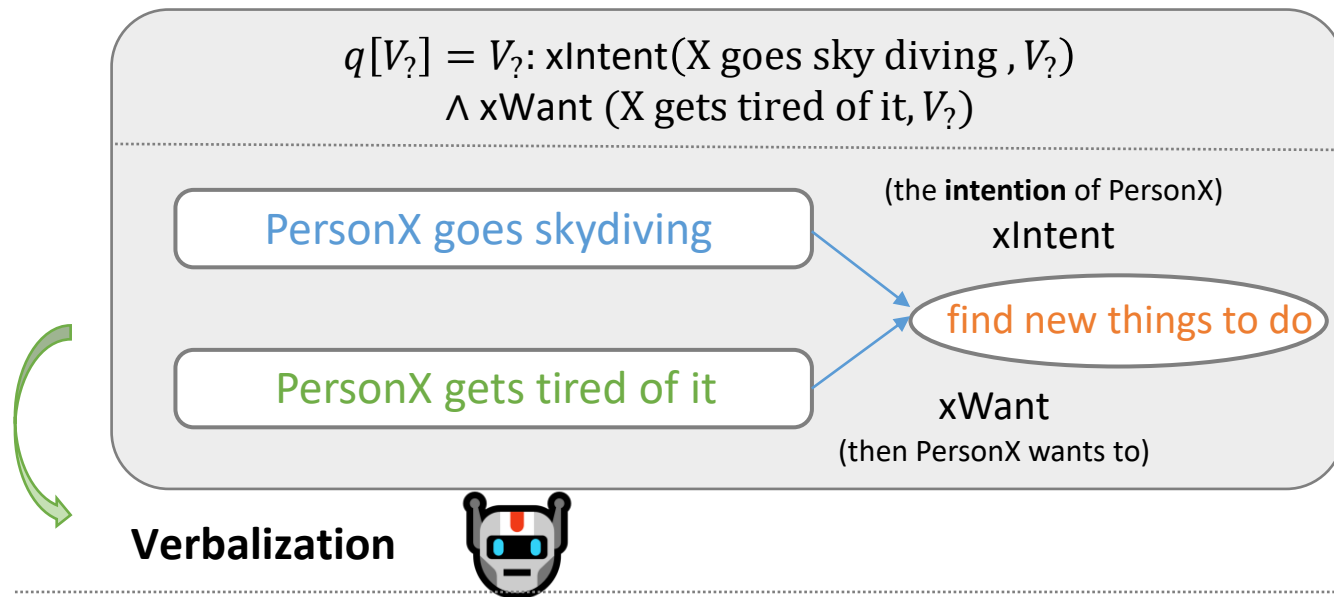
multi-hop effects

Complex Commonsense Reasoning

- Sampling conjunctive logical queries over existing CSKG

- Sampling synthetic query at scale
- Perform verbalization to make the query in natural language
- Define reasoning (question) based on the relations.

Base CSKG: ATOMIC2020



LLM-added
context

PersonX is living a boring life.

Rule-based
discourse

After getting tired of it
PersonX goes skydiving

Question:

What's both the intention of PersonX going skydiving
and what X wants to do after PersonX getting tired of it?

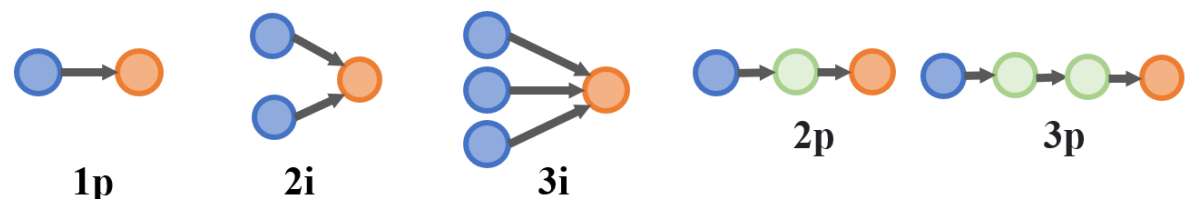
Answer:

find new things to do

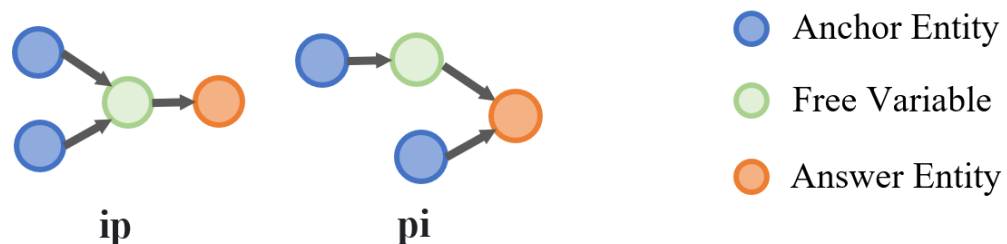
Conjunctive Logical Queries

- Similar to CQA problems

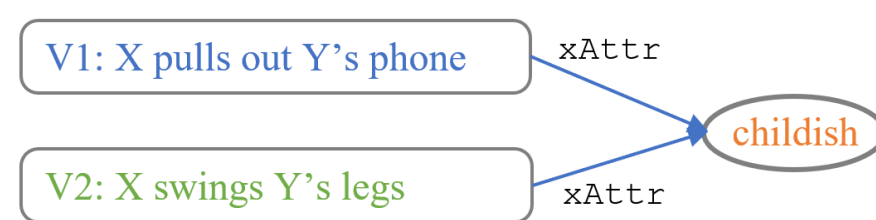
Training Query Types



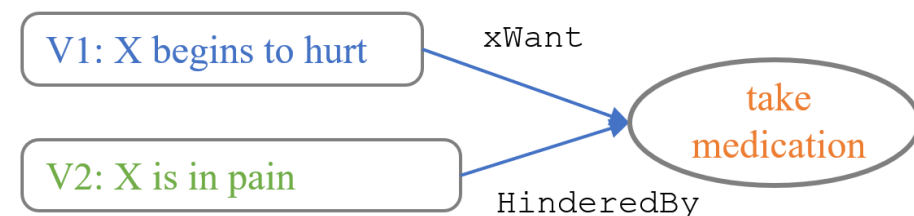
Unseen Query Types



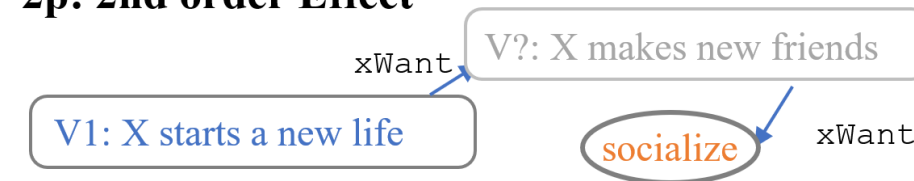
2i: Common Attribution



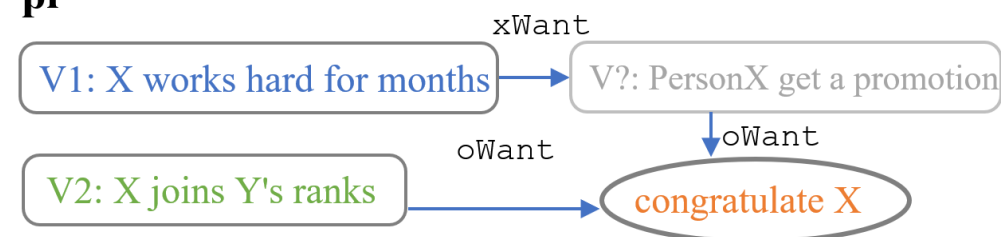
2i-negative: Negated Common Cause



2p: 2nd order Effect



pi



Benchmarking

- Multiple-Choice QA:
 - Negative sampling: 2 are randomly sampled across CSKB, 2 are randomly sampled across one-hop answers (hard negative).
 - An additional “No answers are correct option”
- Generative complex commonsense reasoning
 - 53 annotators
 - Fleiss Kappa: 0.445; IAA: 78%
 - All disagreements are fixed by experts

#Train	790k
#Eval (annotated)	1,317
%wrong verbalization	4.5%
%sampled answers that are plausible	52.1%
%sampled negatives that are plausible	23.5%

LLMs Still Fall Short on This Task

Method	2i	2i-neg	3i	2p	ip	pi	All
API-based LLMs							
gpt-3.5-turbo-0613	33.56	43.12	42.01	38.66	38.05	28.40	37.74
- 1-shot	43.31	35.31	58.45	57.73	51.33	62.96	48.22
- 1-shot w/ CoT	45.80	36.43	54.34	57.73	50.44	66.67	48.75
- 8-shot (2i, 2p)	48.52	41.26	57.08	67.53	53.10	74.07	53.22
- 8-shot (2i, 2p) w/ CoT	52.61	46.10	60.27	59.79	52.21	65.43	54.37
gpt-4-1106-preview	44.67	46.47	52.05	32.47	40.71	53.08	44.64
- 1-shot	47.85	42.01	50.68	38.66	44.25	50.62	45.63
- 1-shot w/ CoT	48.97	46.46	52.96	49.48	52.21	58.02	50.04
- 8-shot (2i, 2p)	54.87	46.47	58.90	45.88	52.21	66.67	53.00
- 8-shot (2i, 2p) w/ CoT	57.82	49.07	62.56	61.34	52.21	66.67	57.40
Open-source (QA) Language Models							
HyKAS (Ma et al., 2021, zero-shot)	34.92	39.41	27.85	41.75	37.17	33.33	35.76
CAR (Wang et al., 2023a, zero-shot)	37.41	30.48	37.44	57.73	32.74	53.09	39.56
Llama2 (7B) (Touvron et al., 2023)	35.15	21.93	39.27	35.57	28.32	51.85	33.64
Vera (5B) (Liu et al., 2023)	47.62	27.51	40.18	66.49	52.21	58.02	46.09
UnifiedQA-v2 (Khashabi et al., 2022)	56.23	39.41	62.56	58.76	51.33	62.96	54.21
Flan-T5 (11B) (Chung et al., 2022)	58.28	47.21	65.30	76.29	56.64	79.01	60.97
Fine-tuned on COM²							
DeBERTa-v3-Large (+COM ²)	60.09	58.36	69.41	61.86	59.29	81.48	62.79
CAR-DeBERTa-v3-Large (+COM ²)	61.22	56.13	69.86	68.56	56.64	85.19	63.78

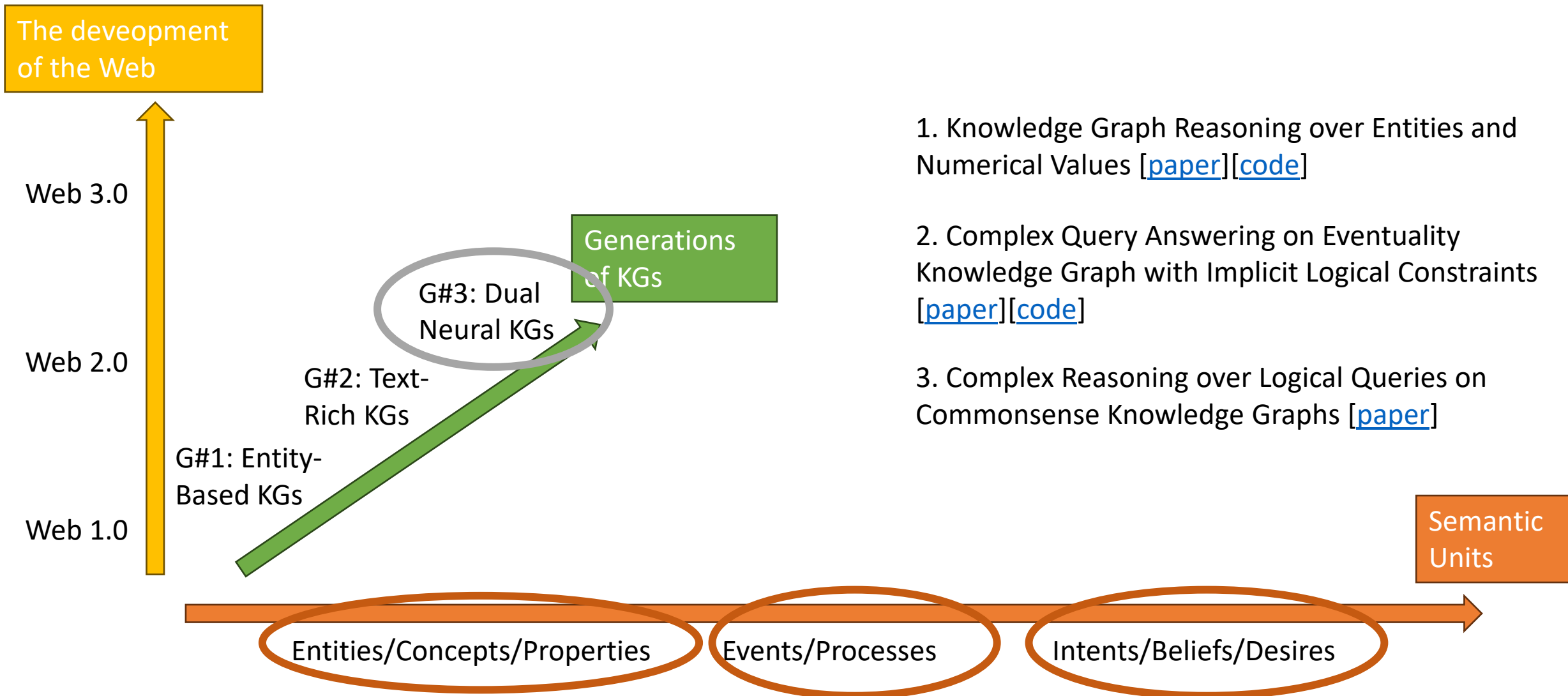
GPT:
37.74%~57.40%

Open LLMs:
33.64%~60.97%

Finetuned:
62.79%~63.78%

Table 1: Model performance (%) on the multiple-choice question answering evaluation set of COM².

The Three-dimensional Development of Knowledge Graphs in Computer Science



Conclusions for Part 4-2

- We have reviewed the frontier of recent development of knowledge graphs in terms methods of different semantic units:
 - Entity/attributes
 - Events/processes
 - Intent/desire/belief
- More recent advances in KG reasoning combined with LLMs will be introduced in Part 5 by Lihui Liu

Roadmap

Part I: Knowledge Graph Reasoning: Basic Concepts and Techniques

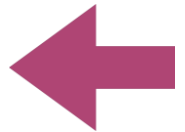
Part II: Recent Advance #1: Neural Reasoning for Natural Language Queries

Part III: Recent Advance #2: Neural Reasoning for Logical Queries

Part IV: Recent Advance #3: Neural Reasoning Beyond Entities and Relations

Part V: Recent Advance #4: LLM+KGR

Part VI: Open Challenges and Future Directions



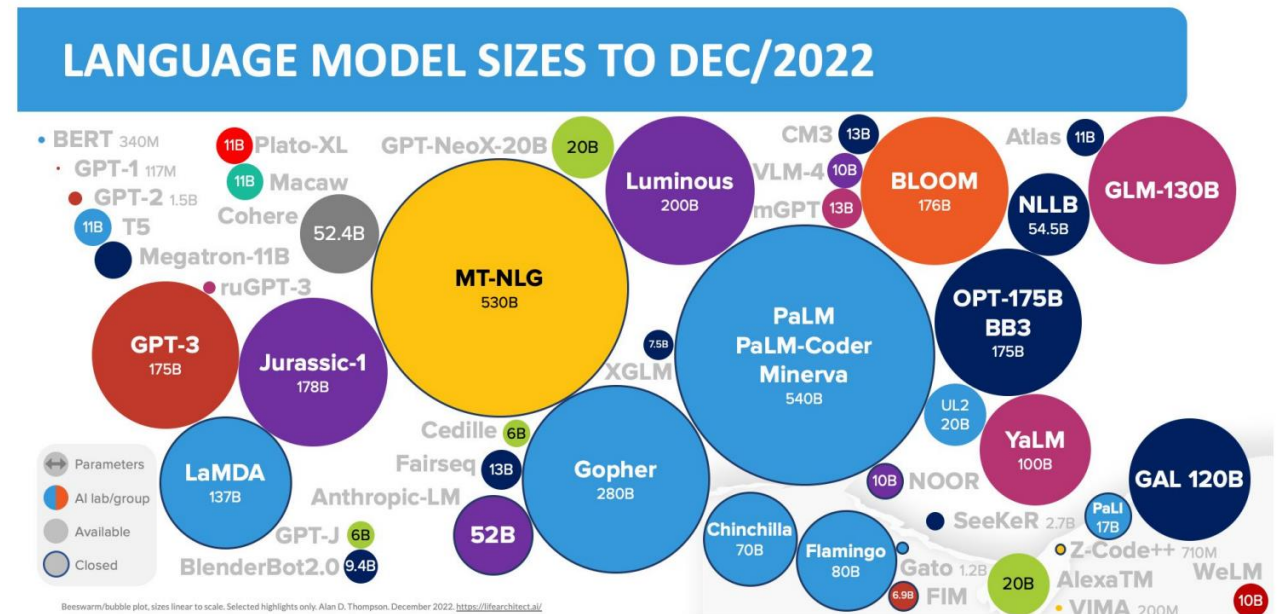
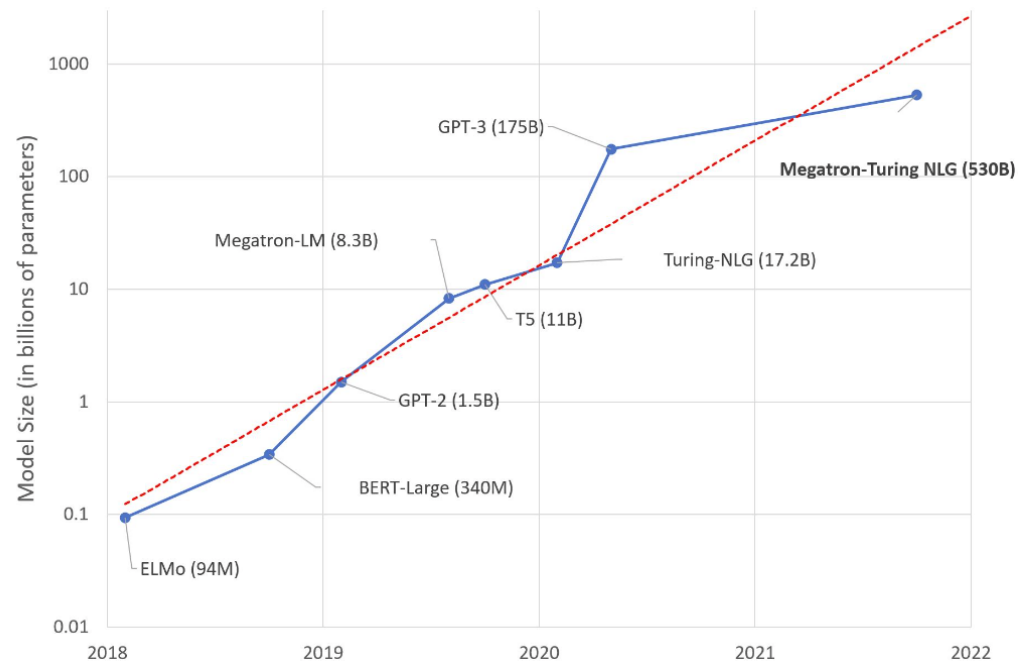
Large Language Models

- ❑ Language model: a probability distribution over strings of text
 - Sally fed my cat with meat: $P(\text{Sally, feed, my, cat, with, meat}) = 0.03$,
 - My cat fed Sally with meat: $P(\text{My, cat, fed, Sally, with, meat}) = 0.005$,
 - Fed cat meat my my with: $P(\text{Fed, cat, meat, my, my, with}) = 0.0001$
- ❑ Large language models (LLM): large, general-purpose language models can be pre-trained and then fine-tuned for specific purposes [2]
 - Examples: GPT-3.5, GPT-4, Llama2 and Llama3

- [1] J. Huang. https://teapot123.github.io/files/CSE561A_slides/Lecture_1.pdf
- [2] Google Cloud Tech. Introduction to Large Language Models. (2023).

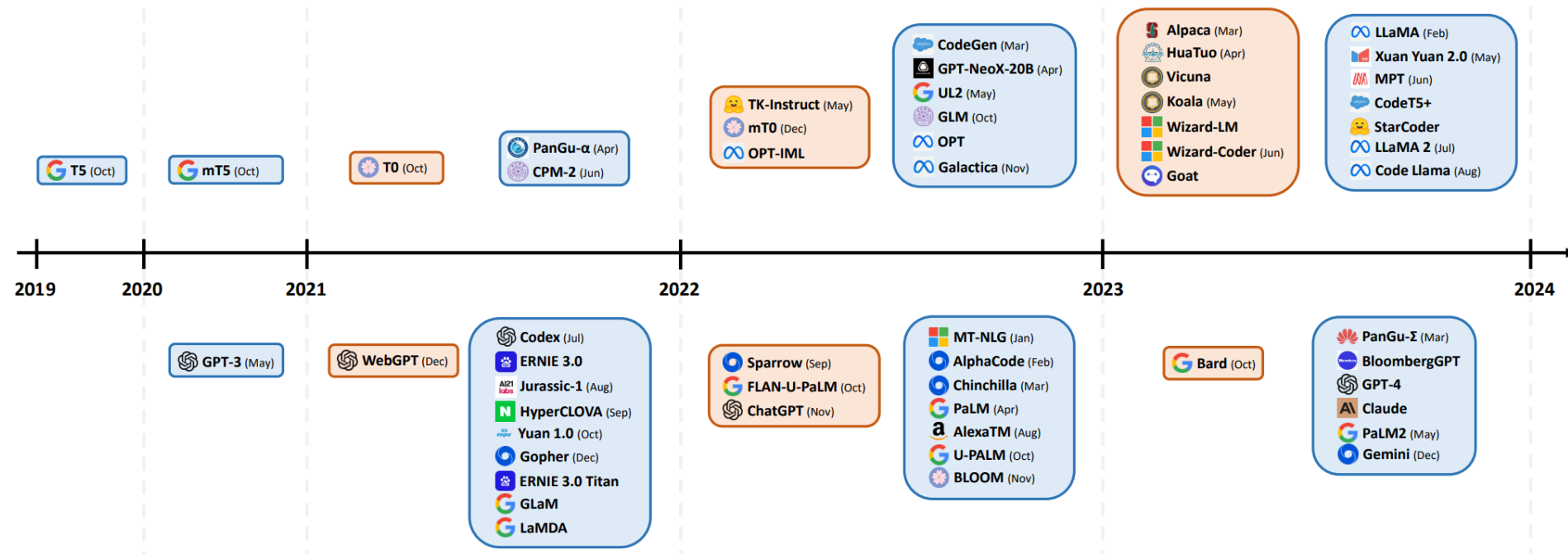
Model Size of LLMs

Size: from millions of parameters to billions of parameters



Large Language Models in Recent Literature

Progression and improvement of large language models



- [1] H. Naveeda, A. Khana, S. Qiub, M. Saqib. 2024. A Comprehensive Overview of Large Language Models (arXiv 24).

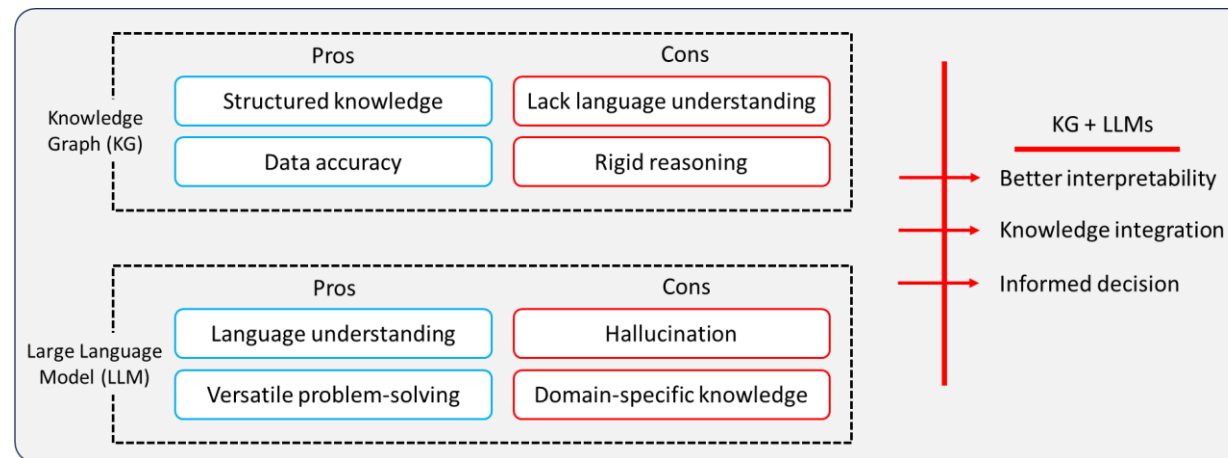
Knowledge Graphs VS LLMs

Knowledge graph

- Pros: accurate structural knowledge, interpretable, ...
- Cons: incomplete, lack language understanding

LLMs

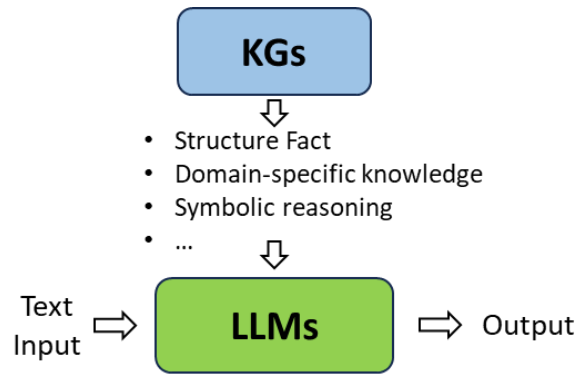
- Pros: general knowledge, good at language understand,...
- Cons: hallucination, lack interpretation, lacking new knowledge, ...



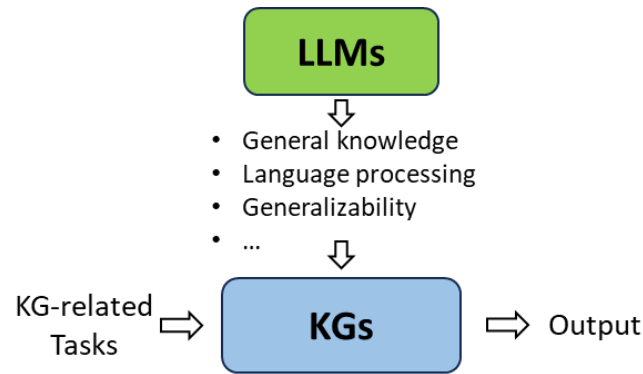
Combining Knowledge Graph with LLMs

□ Categorization

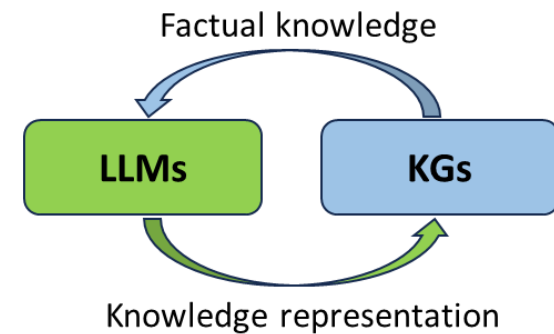
- Knowledge graph enhance LLMs
- LLMs enhance knowledge graph reasoning
- Integrating knowledge graph reasoning with LLMs in a mutually beneficial way



KG-enhanced LLMs



LLMs-augmented KGs



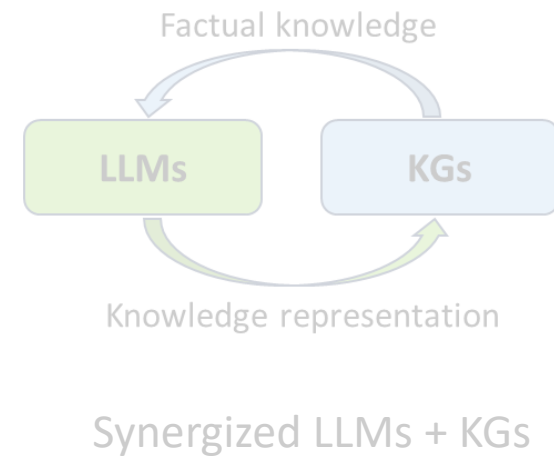
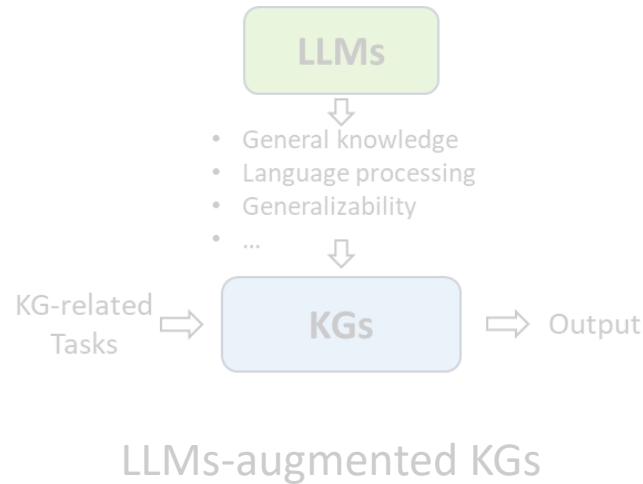
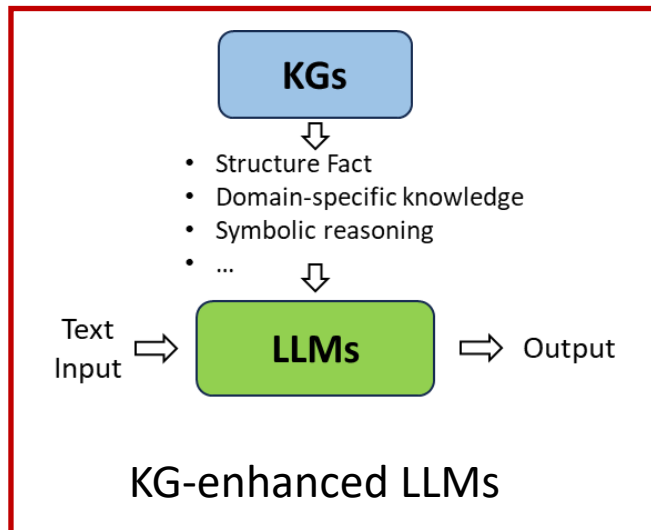
Synergized LLMs + KGs

- [1] S. Pan, L. Luo, Y. Wang, C. Chen. 2024. Unifying Large Language Models and Knowledge Graphs: A Roadmap (TKDE 24).

Combining Knowledge Graph with LLMs

□ Categorization

- Knowledge graph enhance LLMs
- LLMs enhance knowledge graph reasoning
- Integrating knowledge graph reasoning with LLMs in a mutually beneficial way



- [1] S. Pan, L. Luo, Y. Wang, C. Chen. 2024. Unifying Large Language Models and Knowledge Graphs: A Roadmap (TKDE 24).

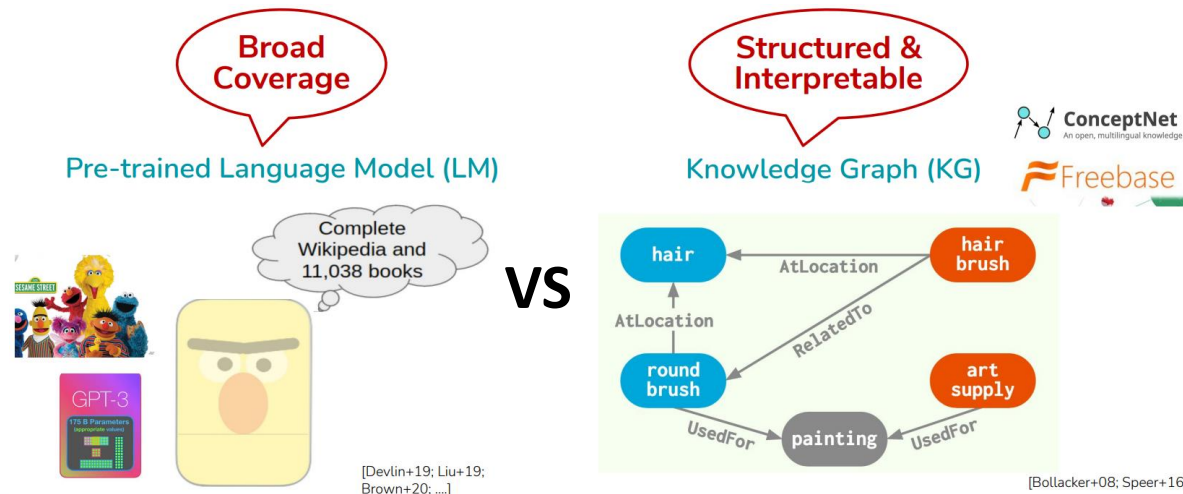
QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering

❑ Goal: answer multi-choice question

If it is not used for **hair**, a **round brush** is an example of what?

A. **hair brush** B. **bathroom** C. **art supplies*** D. **shower**

❑ Philosophy: the system needs to access a lot of knowledge and reason about it



- [1] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, J. Leskovec. 2021. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering (NAACL 21).

QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering

Existing problem:

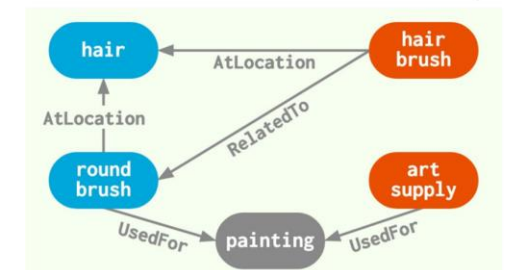
- Language models do not work well for interpretable or logical reasoning; they lack interpretation
- KG is incomplete and noisy

If it is not used for **hair**, a **round brush** is an example of what?
 A. **hair brush** B. **bathroom** C. **art supplies*** D. **shower**

Challenges

- how to identify the relevant information in the knowledge graph
- how to jointly reason over the text and the knowledge graph

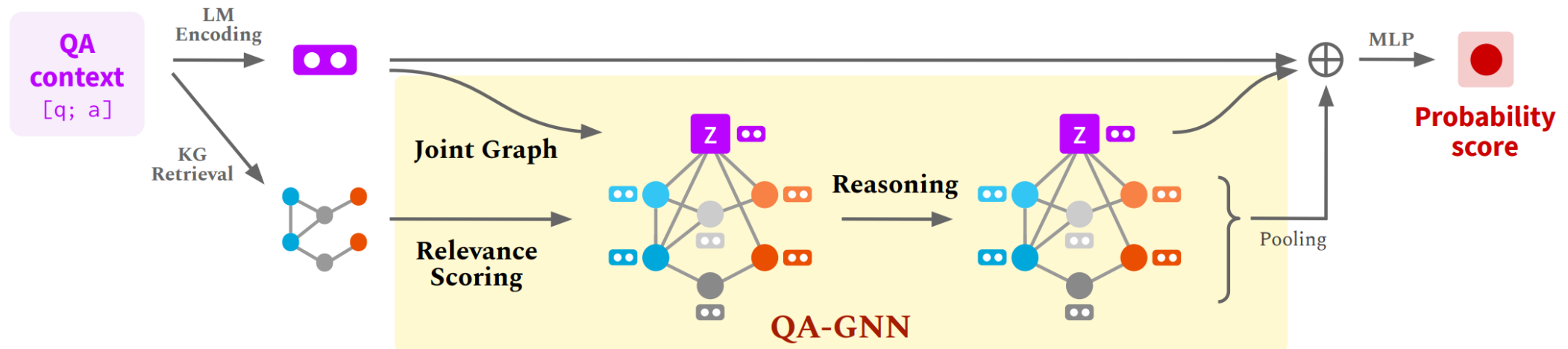
Knowledge Graph (KG)



QA-GNN: Main Steps

□ Idea:

- Language-conditioned KG node relevance scoring
- Joint Reasoning:
 - Connect text and KG to form a joint graph (working graph)
 - Mutually update their representations via Graph Neural Network (GNN)

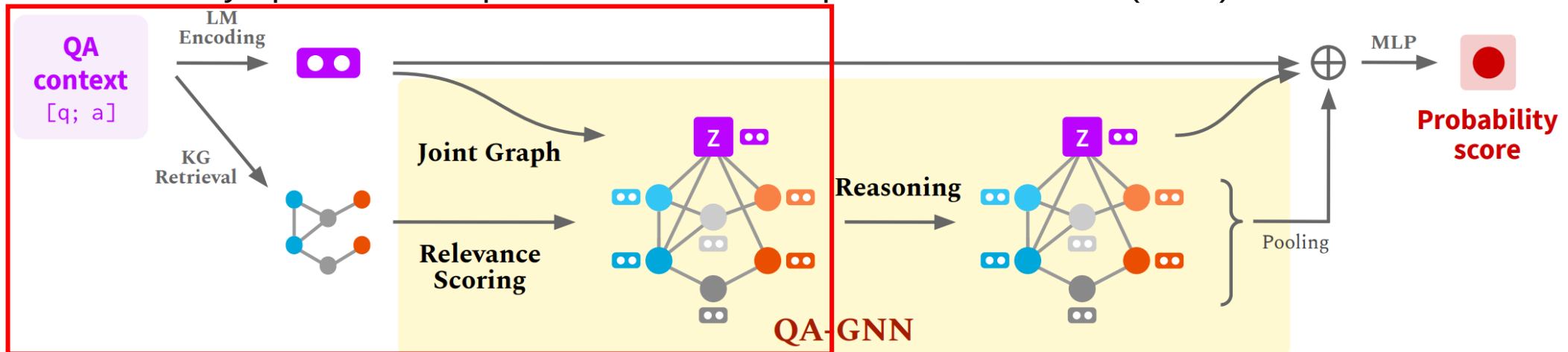


• [1] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, J. Leskovec. 2021. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering (NAACL 21).

QA-GNN: Main Steps

□ Idea:

- Language-conditioned KG node relevance scoring
- Joint Reasoning:
 - Connect text and KG to form a joint graph (working graph)
 - Mutually update their representations via Graph Neural Network (GNN)

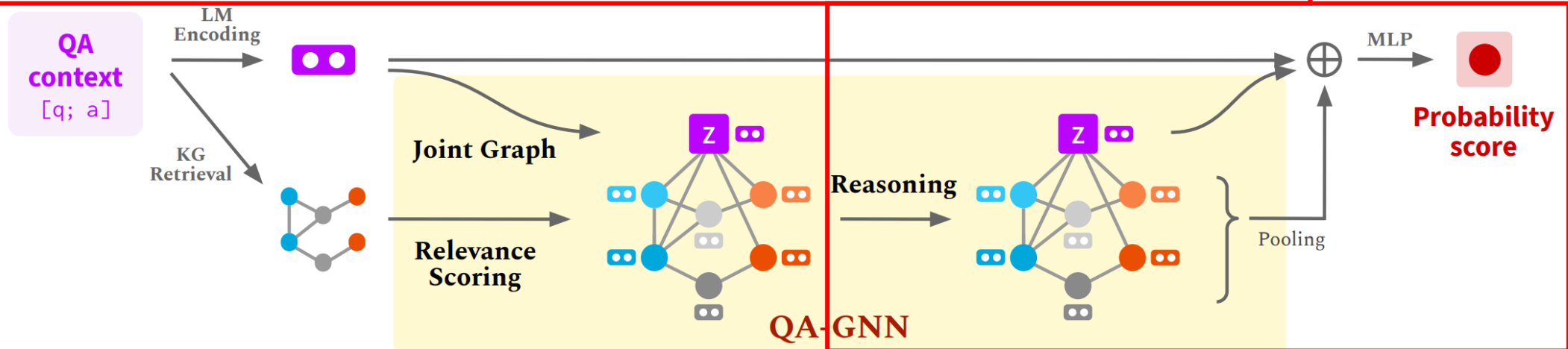


- [1] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, J. Leskovec. 2021. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering (NAACL 21).

QA-GNN: Main Steps

□ Idea:

- Language-conditioned KG node relevance scoring
- Joint Reasoning with language models:
 - Connect text and KG to form a joint graph (working graph)
 - Mutually update their representations via Graph Neural Network (GNN)

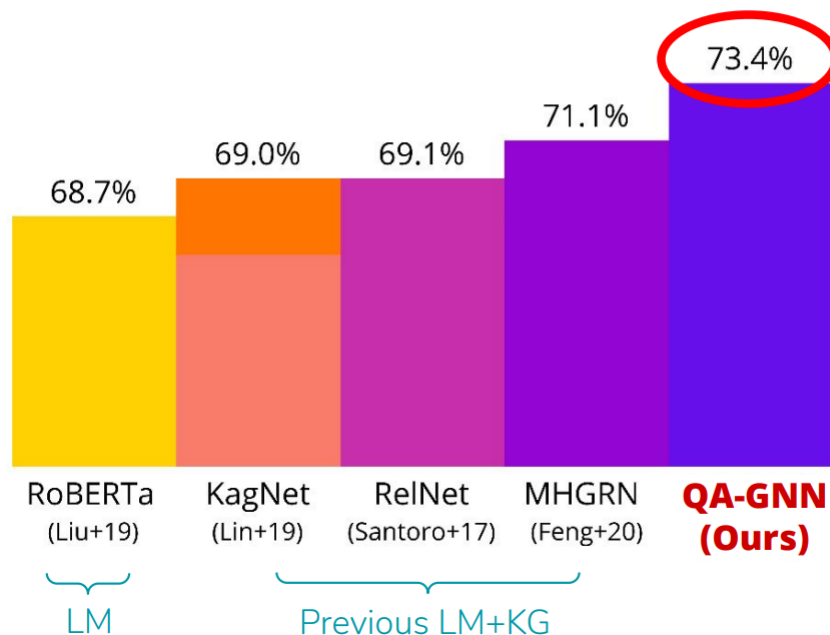


- [1] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, J. Leskovec. 2021. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering (NAACL 21).

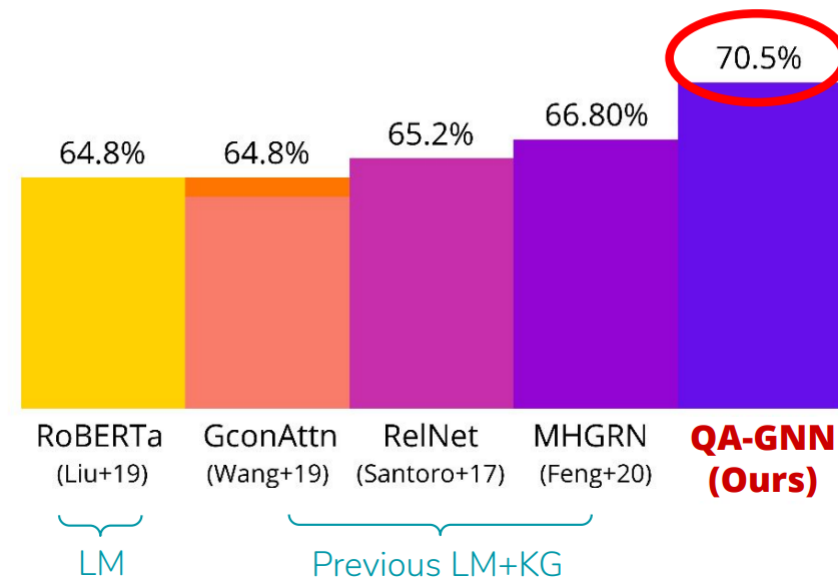
QA-GNN: Experiment Results

Improved performance on two QA tasks

- **CommonsenseQA**: reasoning with commonsense knowledge



- **OpenBookQA**: reasoning with elementary science knowledge

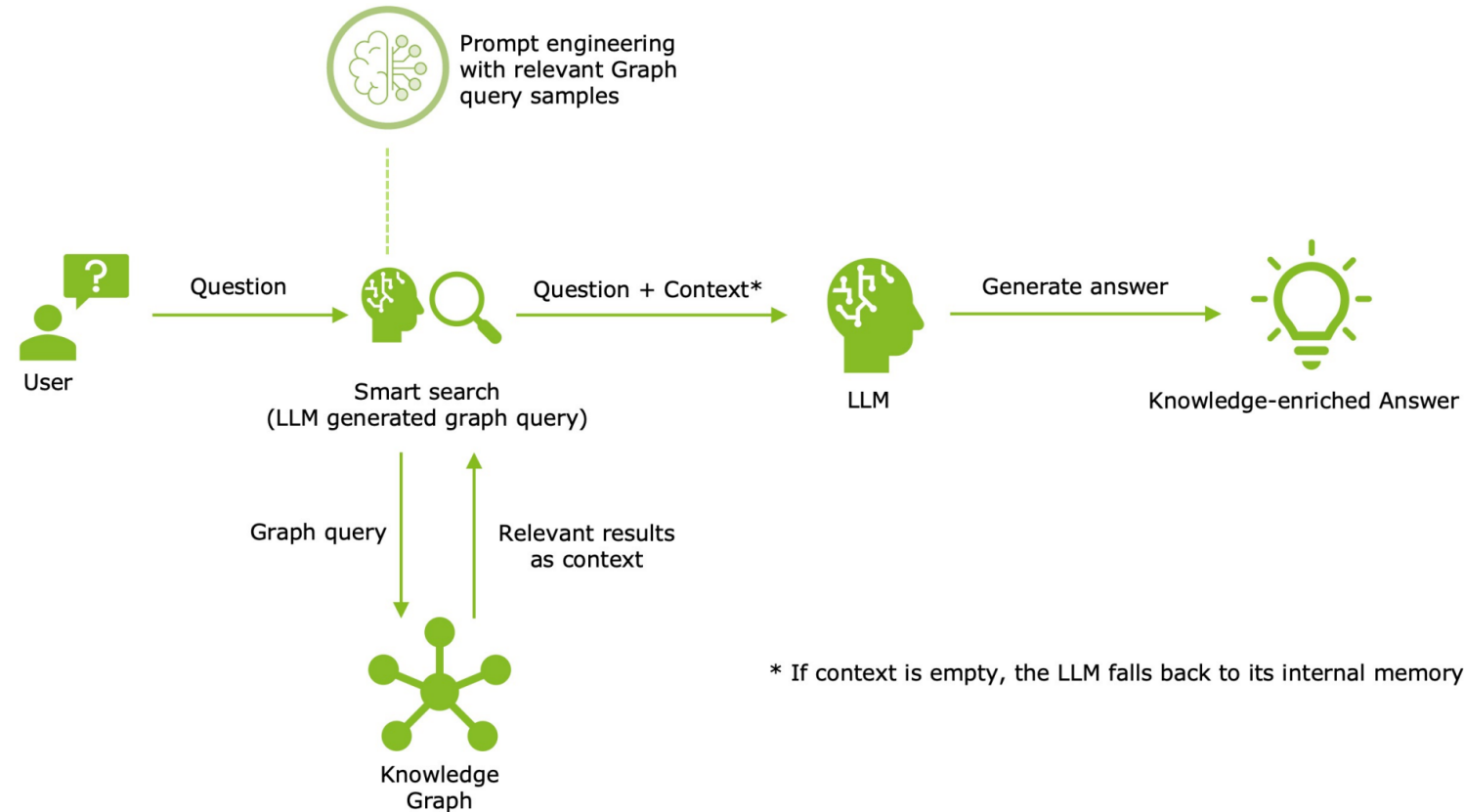


- [1] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, J. Leskovec. 2021. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering (NAACL 21).

Retrieval-Augmented Generation (RAG) with Knowledge Graph

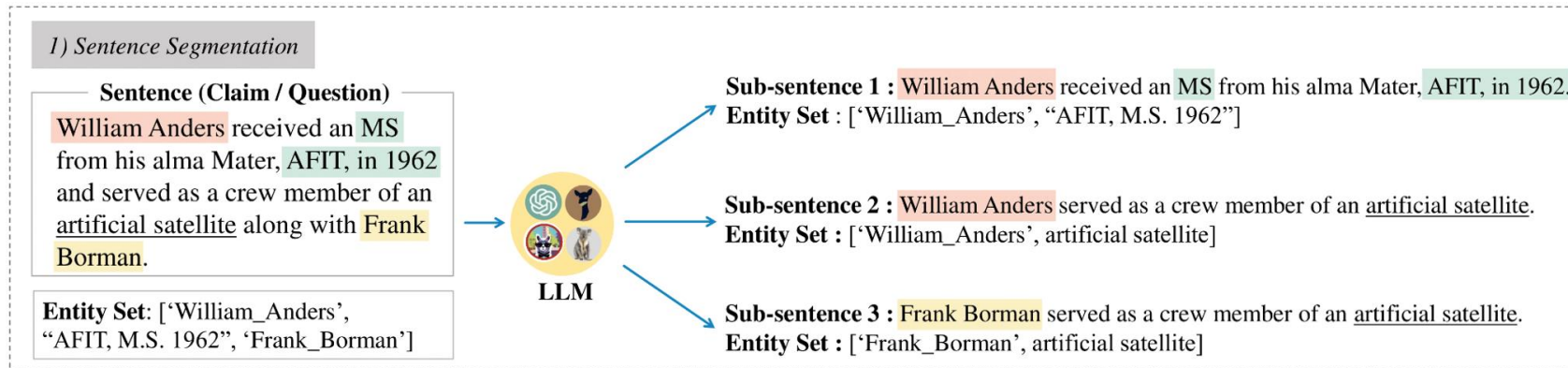
□ Observation:

- When the size of the model becomes large, retrain or finetune the model will be very time consuming



KG-GPT: A General Framework for Reasoning on KGs Using LLMs

- ❑ Goal: utilize language models and knowledge graphs to answer a more complex natural language questions
- ❑ Idea:



KG-GPT: A General Framework for Reasoning on KGs Using LLMs

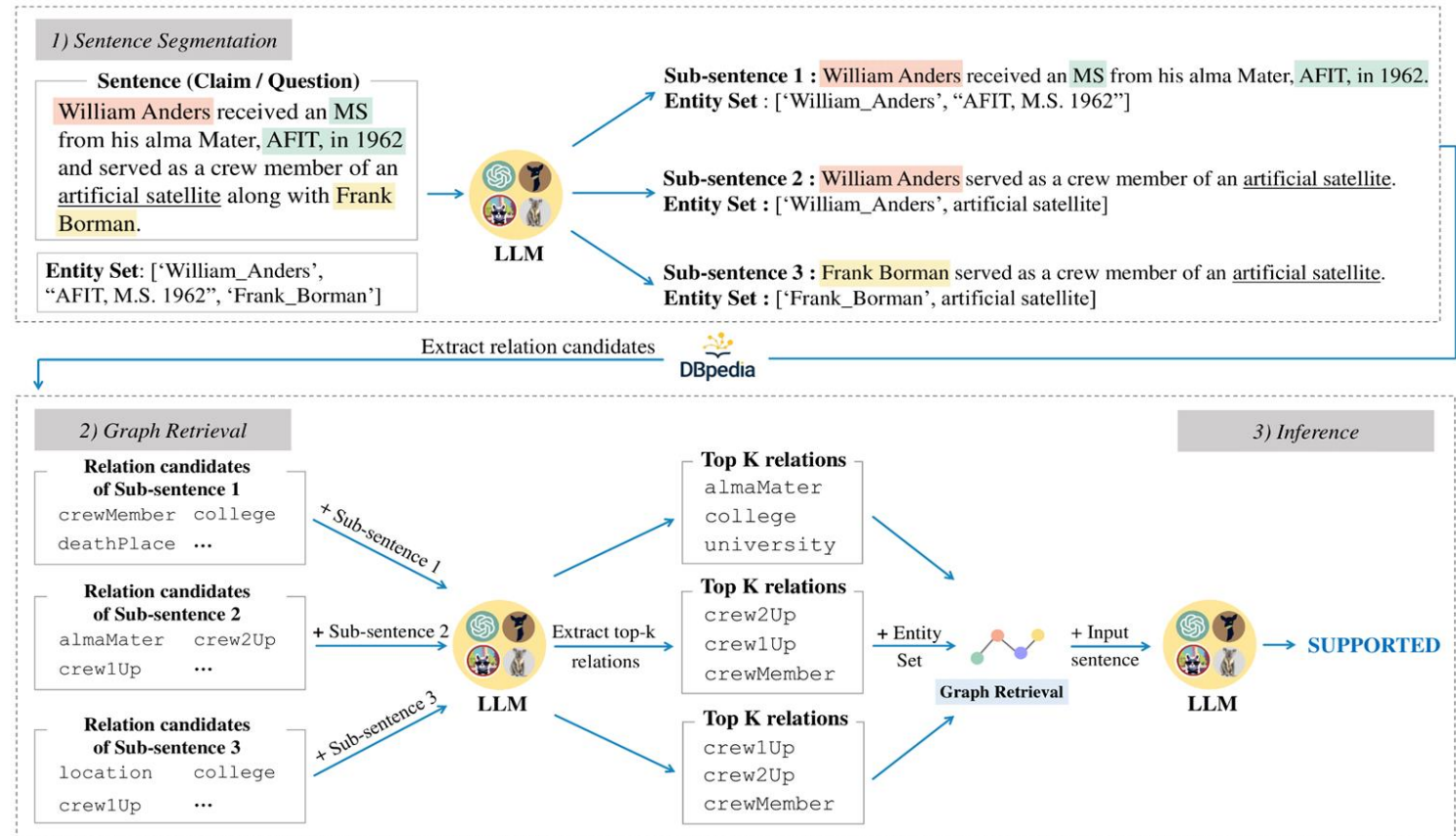
❑ Sentence Segmentation

❑ Graph Retrieval

- Extract relation candidates from KG

❑ Inference

- Graph to prompt



- [1] J. Kim, Y. Kwon, Y. Jo, E. Choi. 2023. KG-GPT: A General Framework for Reasoning on Knowledge Graphs Using Large Language Models (EMNLP 23).

KG-GPT: A General Framework for Reasoning on KGs Using LLMs

- ❑ Observation: combining KG with LLMs can significantly improve accuracy
- ❑ Dataset: MetaQA

Training Strategy	Methods	MetaQA 1-hop	MetaQA 2-hop	MetaQA 3-hop
	KV-Mem	96.2	82.7	48.9
	GraftNet	97.0	94.8	77.7
<i>full</i>	EmbedKGQA	97.5	98.8	94.8
	NSM	97.1	99.9	98.9
	UniKGQA	97.5	99.0	99.1
➔ <i>12-shot</i>	ChatGPT	60.0	23.0	38.7
<i>12-shot</i>		96.3	94.4	94.0
<i>8-shot</i>	KG-GPT	95.8	93.8	68.8
<i>4-shot</i>		94.7	92.8	46.6

- [1] J. Kim, Y. Kwon, Y. Jo, E. Choi. 2023. KG-GPT: A General Framework for Reasoning on Knowledge Graphs Using Large Language Models (EMNLP 23).

REPLUG: Retrieval-Augmented Black-Box Language Models

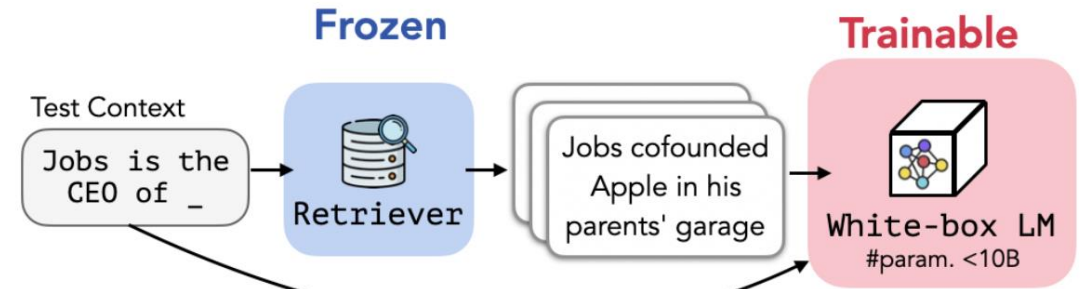
Previous methods:

- Enhance a language model with retrieval by **updating the LM's parameters**

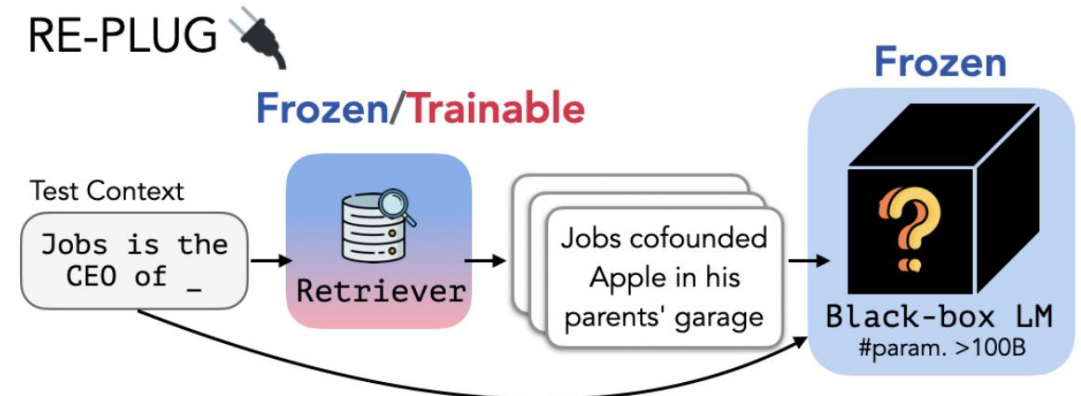
REPLUG:

- Treats the language model as a black box
- Augments it with a frozen or tunable retriever
- Makes REPLUG applicable to large LMs, which are often served via APIs.

Previous

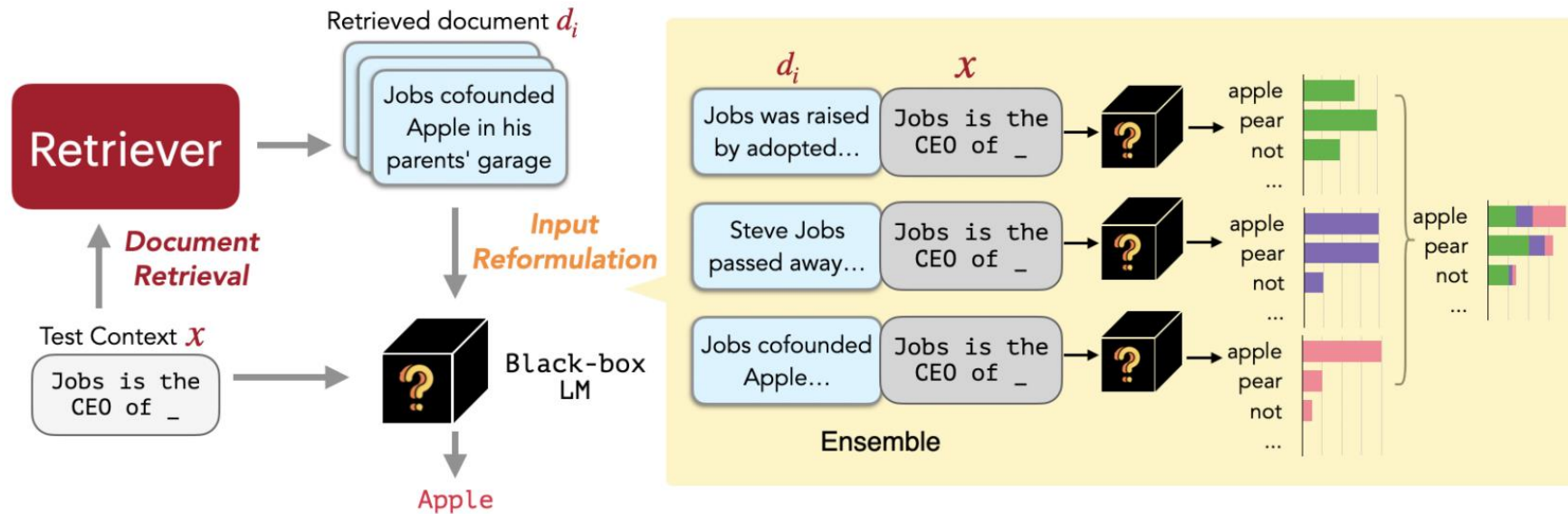


RE-PLUG



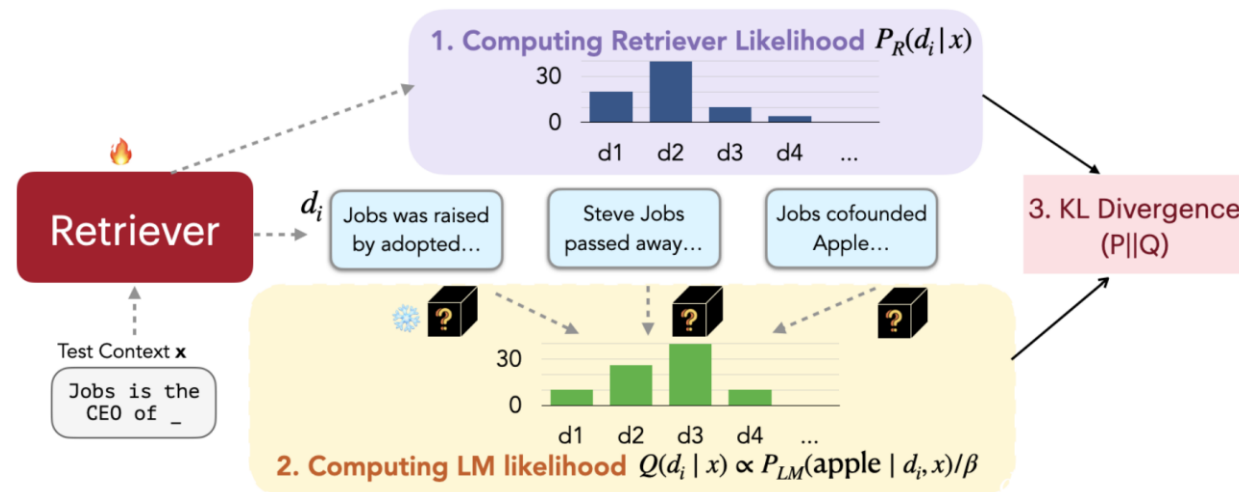
REPLUG

- ❑ First retrieves a small set of relevant documents from an external corpus
- ❑ Then pass the concatenation of each retrieved document with the input context through the LM in parallel, and ensemble the predicted probabilities



REPLUG LSR

- ❑ (1) Using retriever to computing the retrieval likelihood
- ❑ (2) Computing LM likelihood
- ❑ (3) Updating the retrieval model parameters by minimizing the KL divergence between the retrieval likelihood and the LM's score distribution



From Local to Global: A Graph RAG Approach to Query-Focused Summarization

❑ Existing problem

- Fail on global questions such as “what are the main themes in the dataset?”

❑ GraphRAG

- Indexing process aimed at creating LLM Memory Representation on private data
 - Generating Knowledge Graphs
 - Dividing entities to different communities

Preprocessing Pipeline – Graph Enabled RAG

Private Dataset



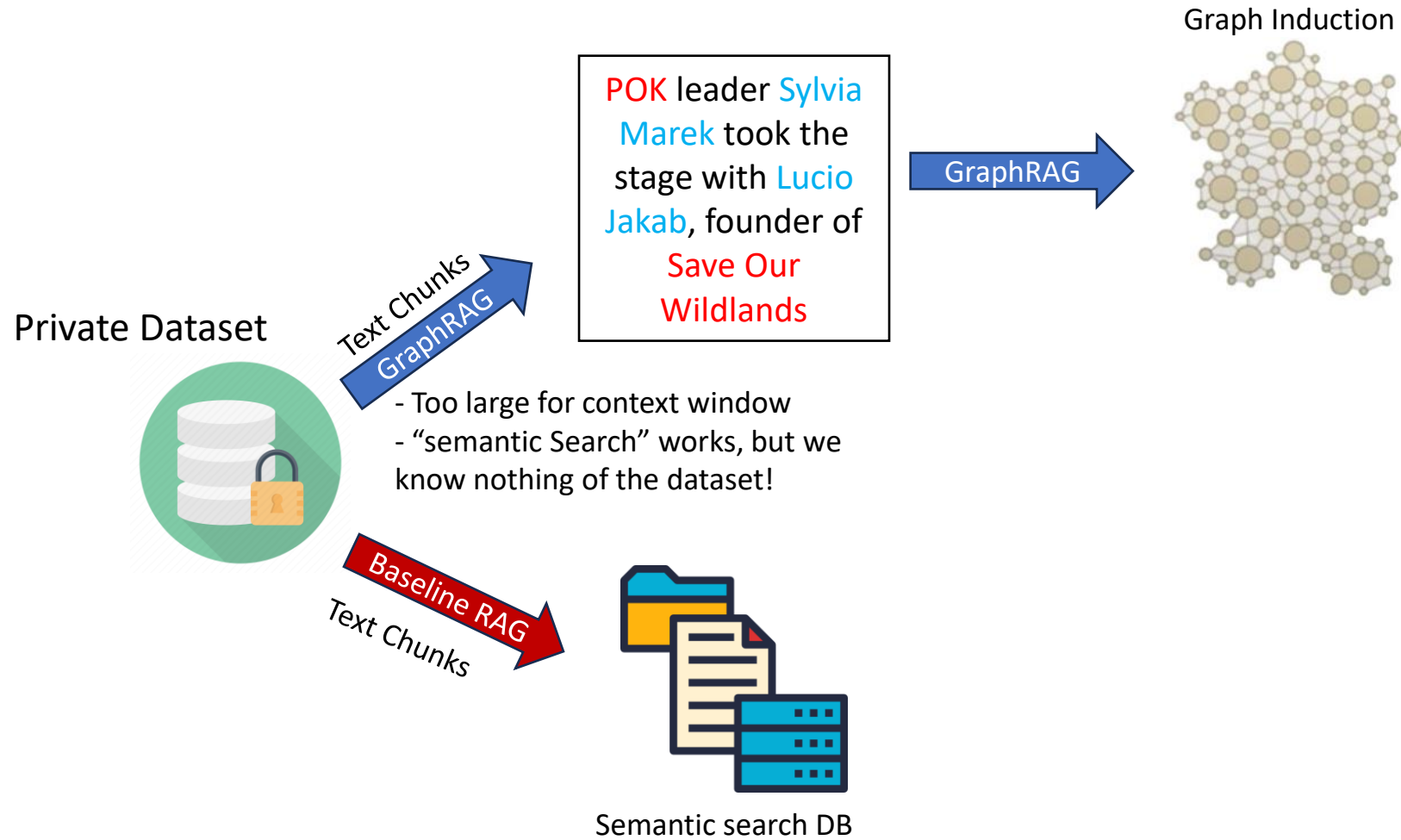
- Too large for context window
- “semantic Search” works, but we know nothing of the dataset!

Baseline RAG
Text Chunks

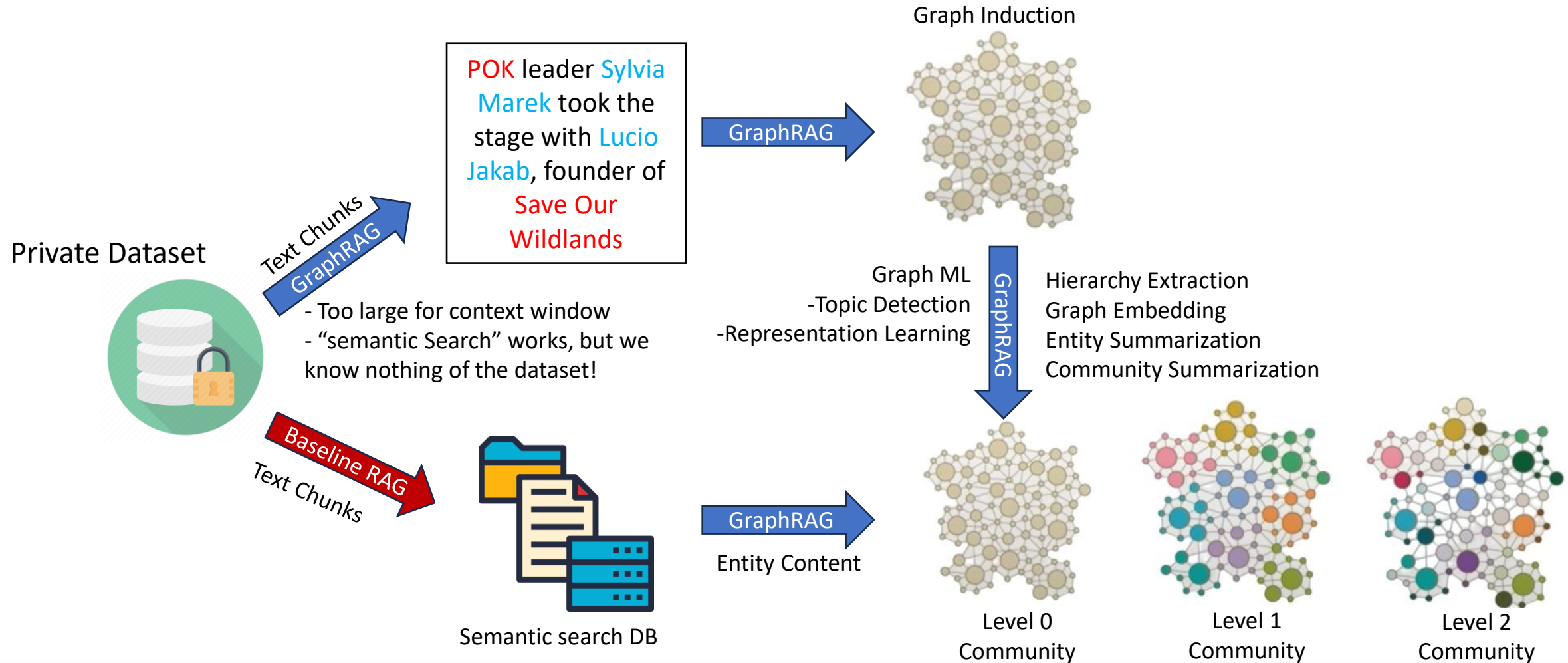


Semantic search DB

Preprocessing Pipeline – Graph Enabled RAG



Preprocessing Pipeline – Graph Enabled RAG

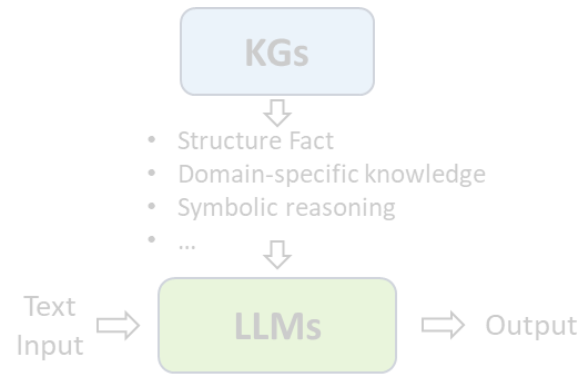




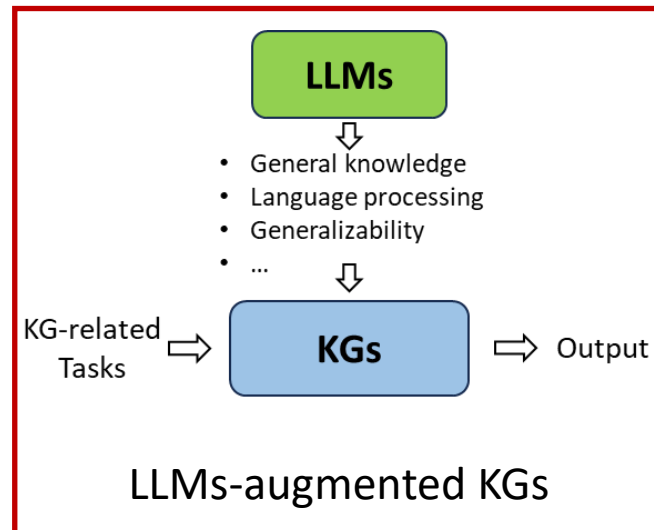
Combining Knowledge Graph with LLMs

□ Categorization

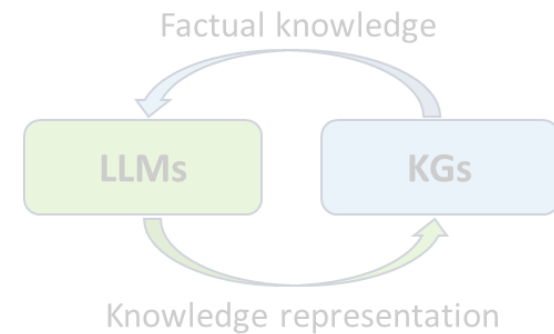
- LLMs enhance knowledge graph reasoning
- Knowledge graph reasoning enhance LLMs
- Integrating knowledge graph reasoning with LLMs in a mutually beneficial way



KG-enhanced LLMs



LLMs-augmented KGs



Synergized LLMs + KGs

- [1] S. Pan, L. Luo, Y. Wang, C Chen. 2024. Unifying Large Language Models and Knowledge Graphs: A Roadmap (TKDE 24).

Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph

- ❑ Traditional methods
 - ❑ Retrieve information from KGs
 - ❑ Augment the prompt and feed it into LLMs
 - ❑ LLMs do not participate in the graph reasoning process directly
- ❑ Existing problem
 - ❑ Suffer from knowledge graph incompleteness

Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph

- Goal: treats the LLM as an agent to travel KGs and perform reasoning based on paths

Question:

What is the majority party now in the country where **Canberra** is located?

LLM-only

Method1 →

(Chain-of-Thought Prompt): Let's think step by step.

Response: **Canberra** is the capital of **Australia**. According to my knowledge up to September 2021 the prime minister of Australia is **Scott Morrison**, who is a member of the **Liberal Party**. So the answer should be **Liberal Party**. ❌

(a)

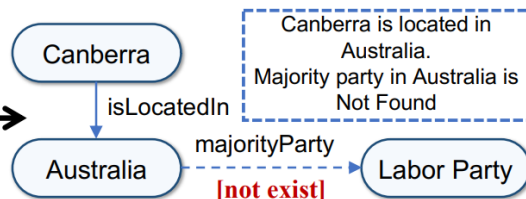
LLM ⊕ KG

Method2 →

(Prompt): Please generate a SPARQL query for this question.

Response: `SELECT ?country ?party WHERE {
 ?canberra dbprop:isLocatedIn ?country .
 ?country dbprop:majorityParty ?party .
}`

Retrieve



Prompt

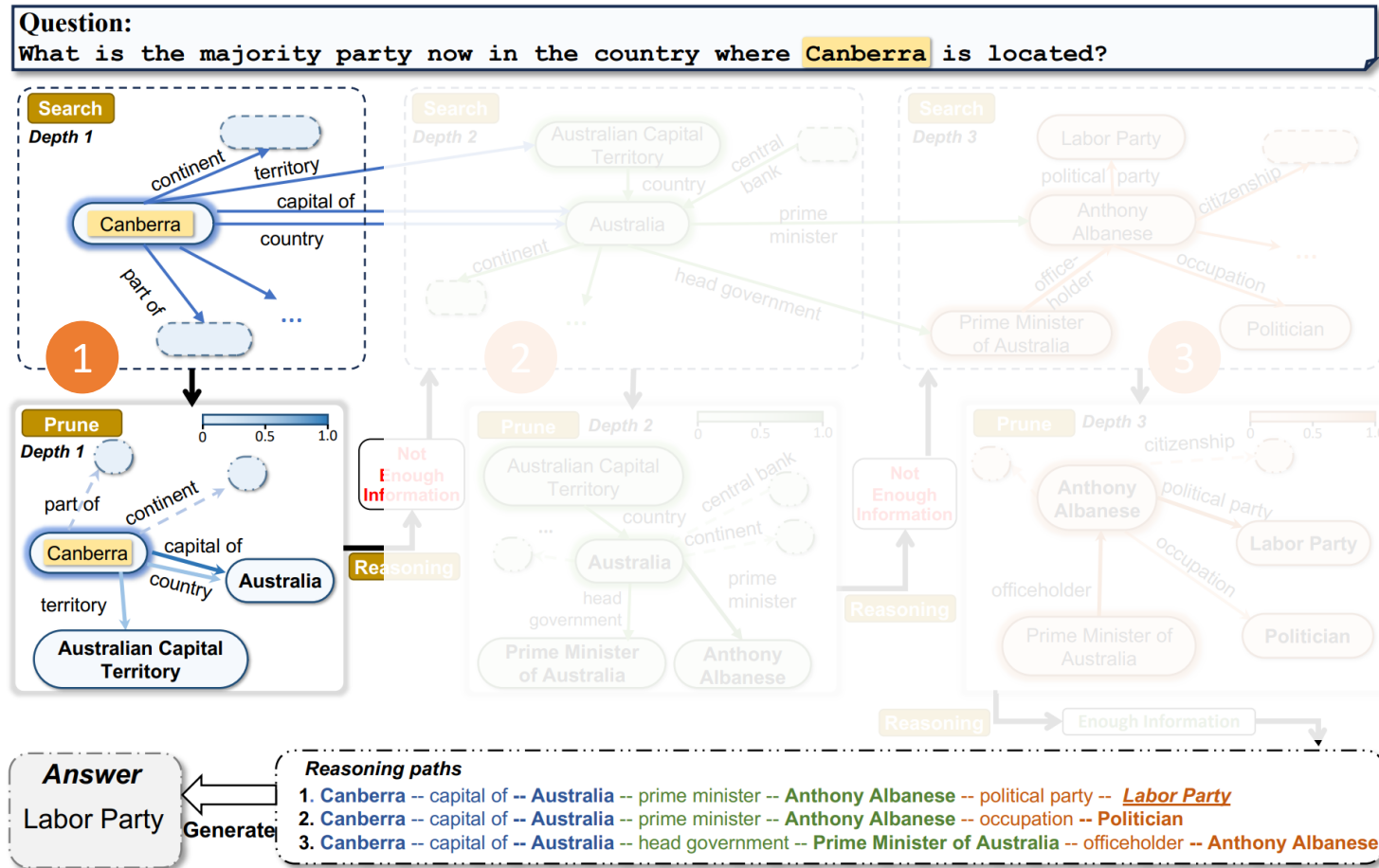
Response: Sorry, based on my query result from the knowledge base, I **cannot answer your question** since I do not have enough information. ❌

(b)

- [1] J. Sun, C. Xu, L. Tang, S. Wang, C. Lin. 2024. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph (ICLR 24).

Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph

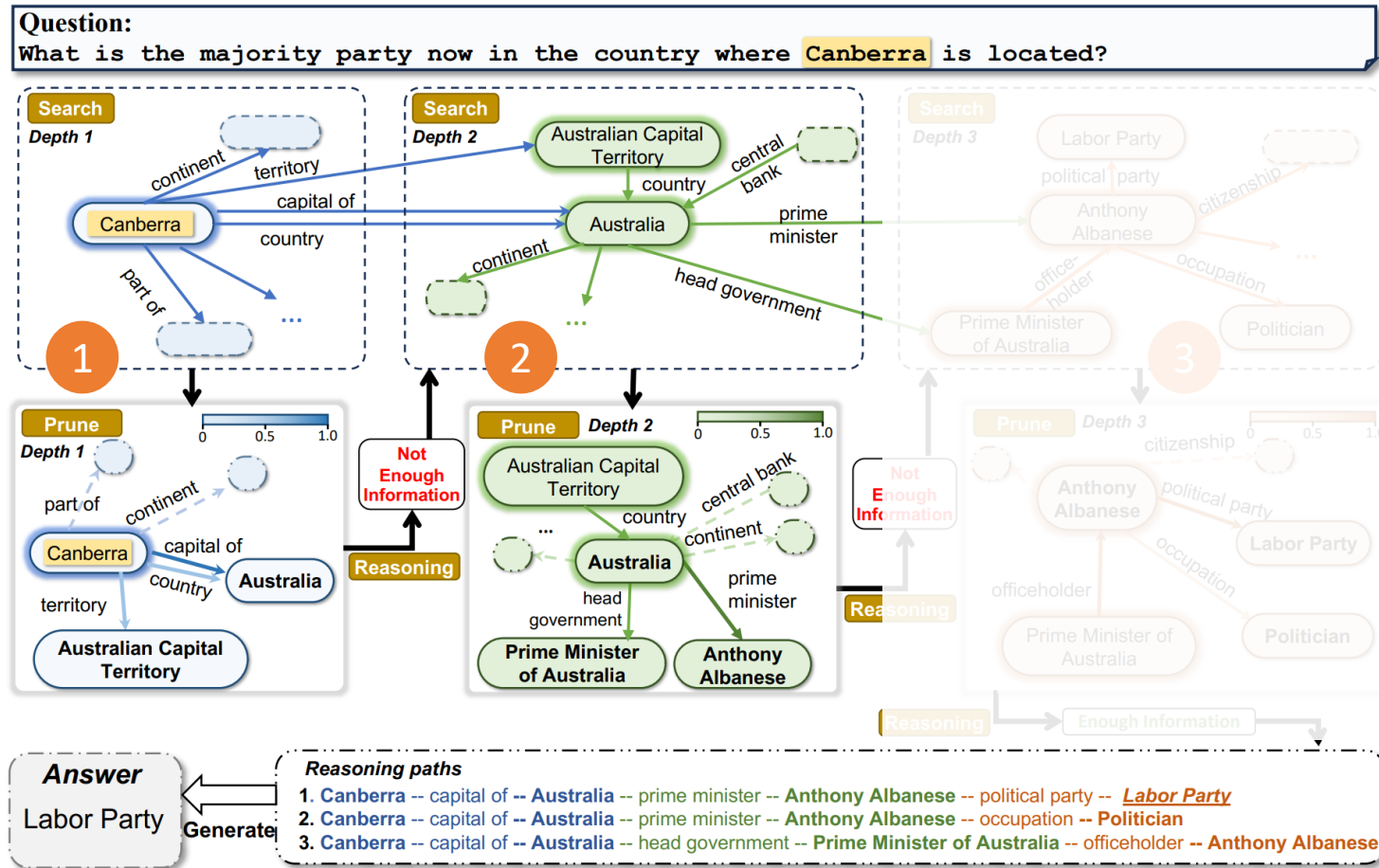
□ Example:



- [1] J. Sun, C. Xu, L. Tang, S. Wang, C. Lin. 2024. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph (ICLR 24).

Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph

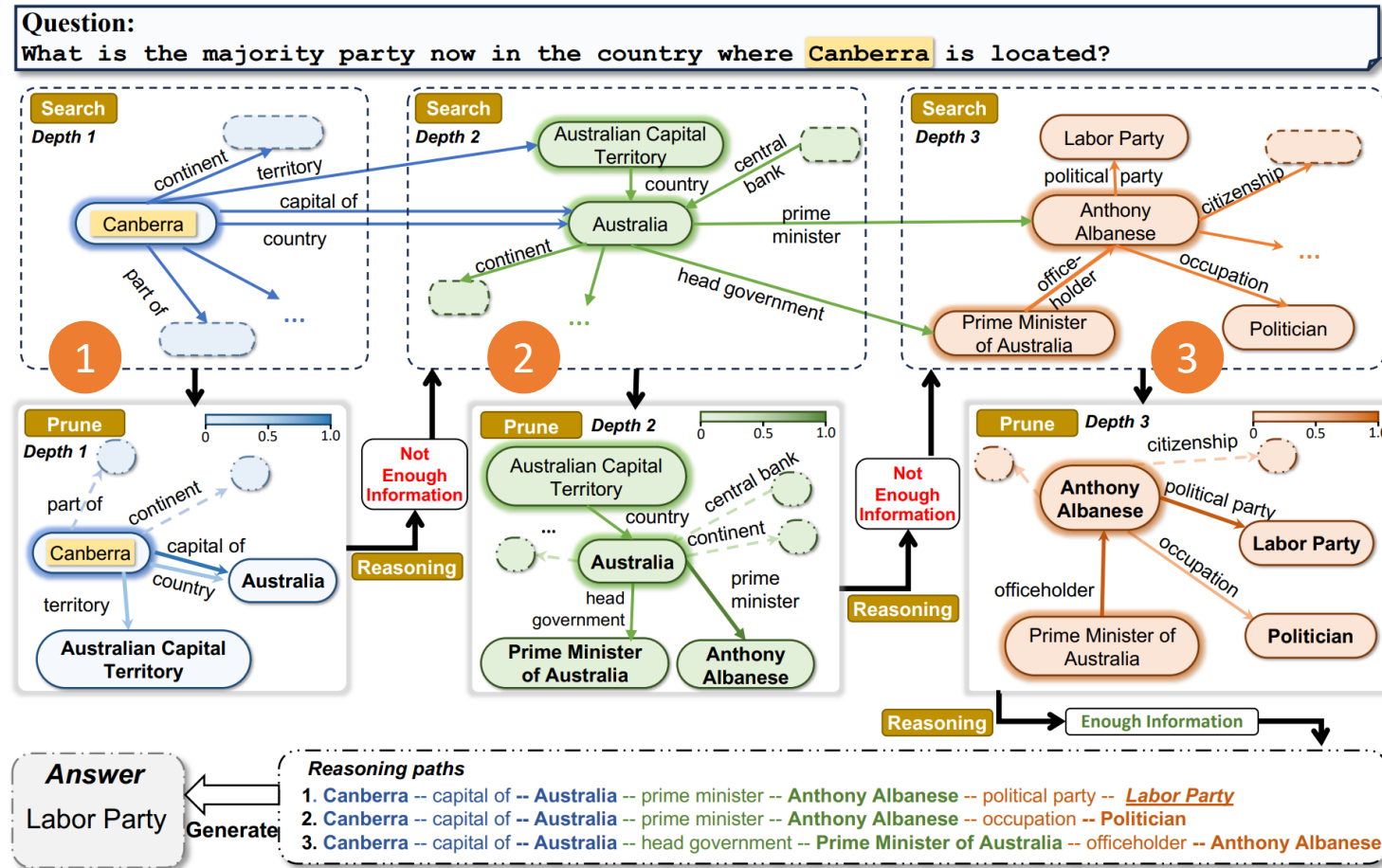
□ Example:



- [1] J. Sun, C. Xu, L. Tang, S. Wang, C. Lin. 2024. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph (ICLR 24).

Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph

□ Example:

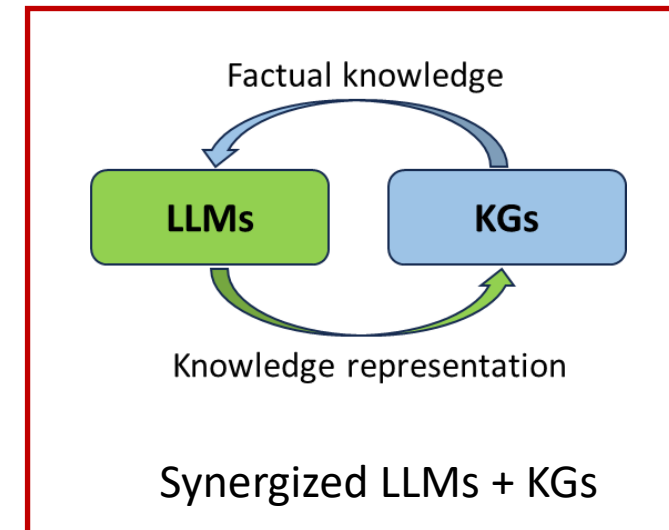
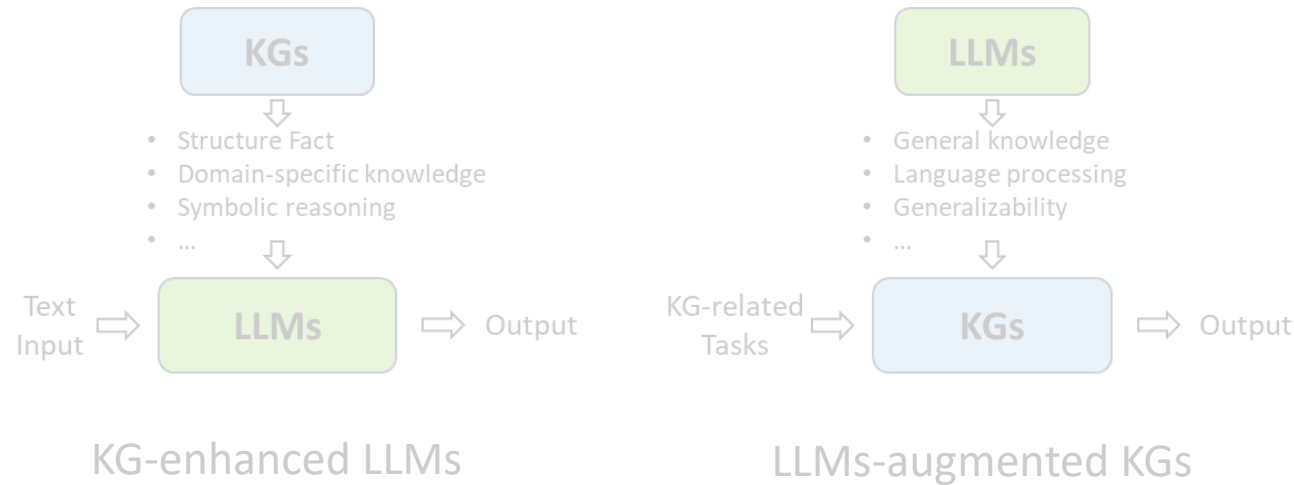


- [1] J. Sun, C. Xu, L. Tang, S. Wang, C. Lin. 2024. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph (ICLR 24).

Combining Knowledge Graph with LLMs

□ Categorization

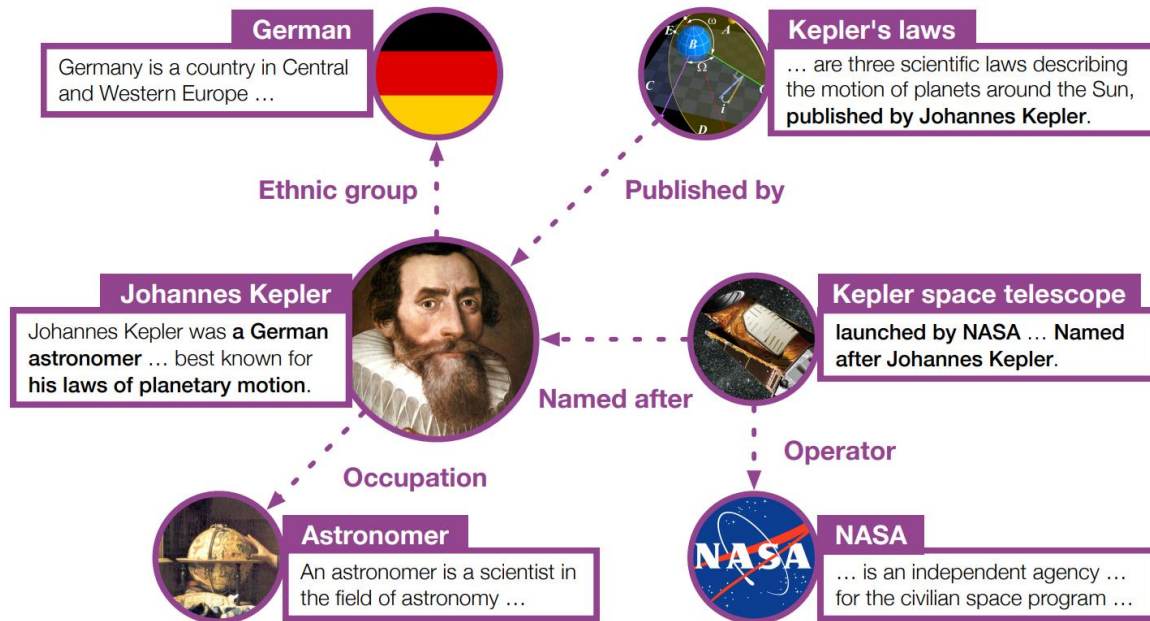
- LLMs enhance knowledge graph reasoning
- Knowledge graph reasoning enhance LLMs
- Integrating knowledge graph reasoning with LLMs in a mutually beneficial way



- [1] S. Pan, L. Luo, Y. Wang, C. Chen. 2024. Unifying Large Language Models and Knowledge Graphs: A Roadmap (TKDE 24).

KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation

□ Motivation:



Knowledge graph embedding



Infuse world facts



Leverage textual information
do inductive KE

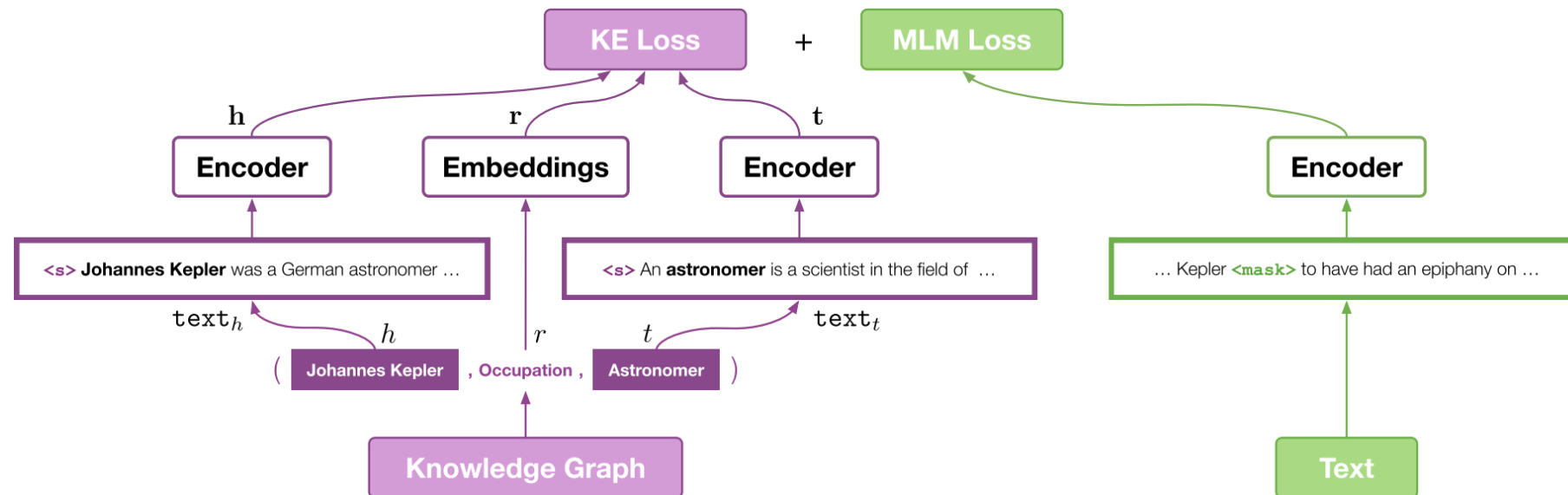


Pre-trained Language Models

- [1] X. Wang, T. Gao, Z. Zhu, Z. Zhang. 2021. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation (ACL 21).

KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation

- Goal: joint optimizing knowledge graph embedding and masked language modeling (MLM) objectives



- [1] X. Wang, T. Gao, Z. Zhu, Z. Zhang. 2021. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation (ACL 21).

KEPLER: Experiments

□ Observations:

- State of the art performance on KE task
- State of the art performance on various NLP tasks

Model	MR	MRR	HITS@1	HITS@3	HITS@10
TransE (Bordes et al., 2013)	109370	25.3	17.0	31.1	39.2
TransE [†]	406957	6.0	1.8	8.0	13.6
DKRL (Xie et al., 2016)	31566	16.0	12.0	18.1	22.9
RoBERTa	1381597	0.1	0.0	0.1	0.3
Our RoBERTa	1756130	0.1	0.0	0.1	0.2
KEPLER-KE	76735	8.2	4.9	8.9	15.1
KEPLER-Rel	15820	6.6	3.7	7.0	11.7
KEPLER-Wiki	14454	15.4	10.5	17.4	24.4
KEPLER-Cond	20267	21.0	17.3	22.4	27.7

Performance on knowledge graph embedding task.

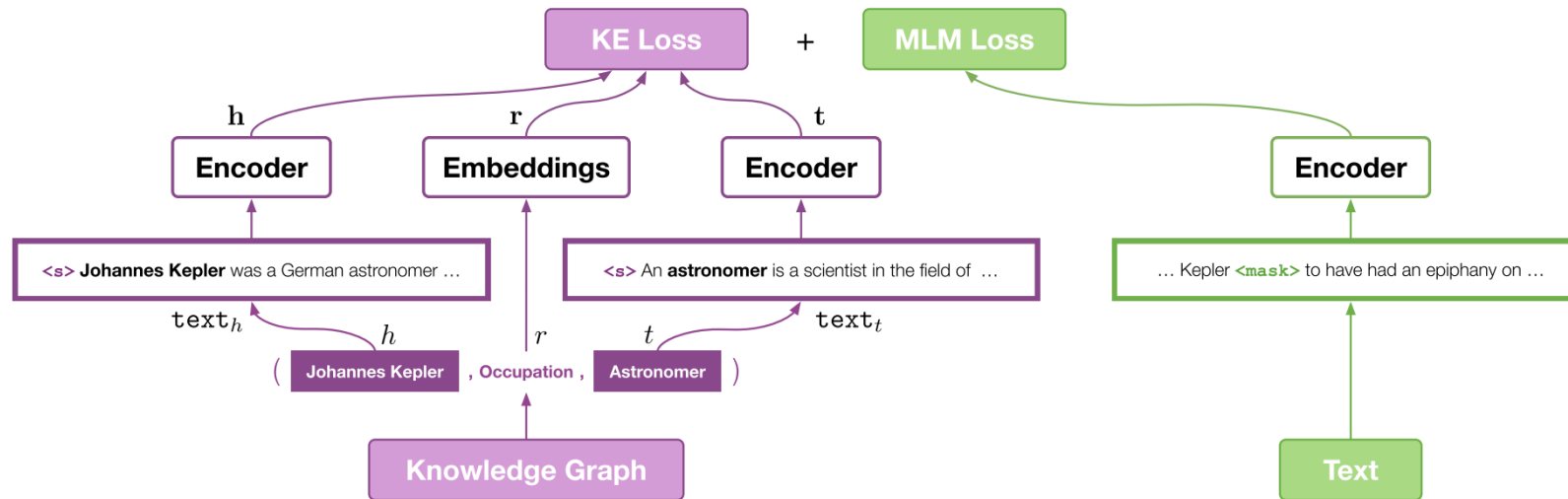
Model	P	R	F-1
UFET (Choi et al., 2018)	77.4	60.6	68.0
BERT	76.4	71.0	73.6
ERNIE _{BERT}	78.4	72.9	75.6
KnowBert _{BERT}	77.9	71.2	74.4
RoBERTa	77.4	73.6	75.4
ERNIE _{RoBERTa}	80.3	70.2	74.9
KnowBert _{RoBERTa}	78.7	72.7	75.6
Our RoBERTa	75.1	73.4	74.3
KEPLER-Wiki	77.8	74.6	76.2

Entity typing results on OpenEntity (%)

- [1] X. Wang, T. Gao, Z. Zhu, Z. Zhang. 2021. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation (ACL 21).

JAKET: Joint Pre-training of Knowledge Graph and Language Understanding

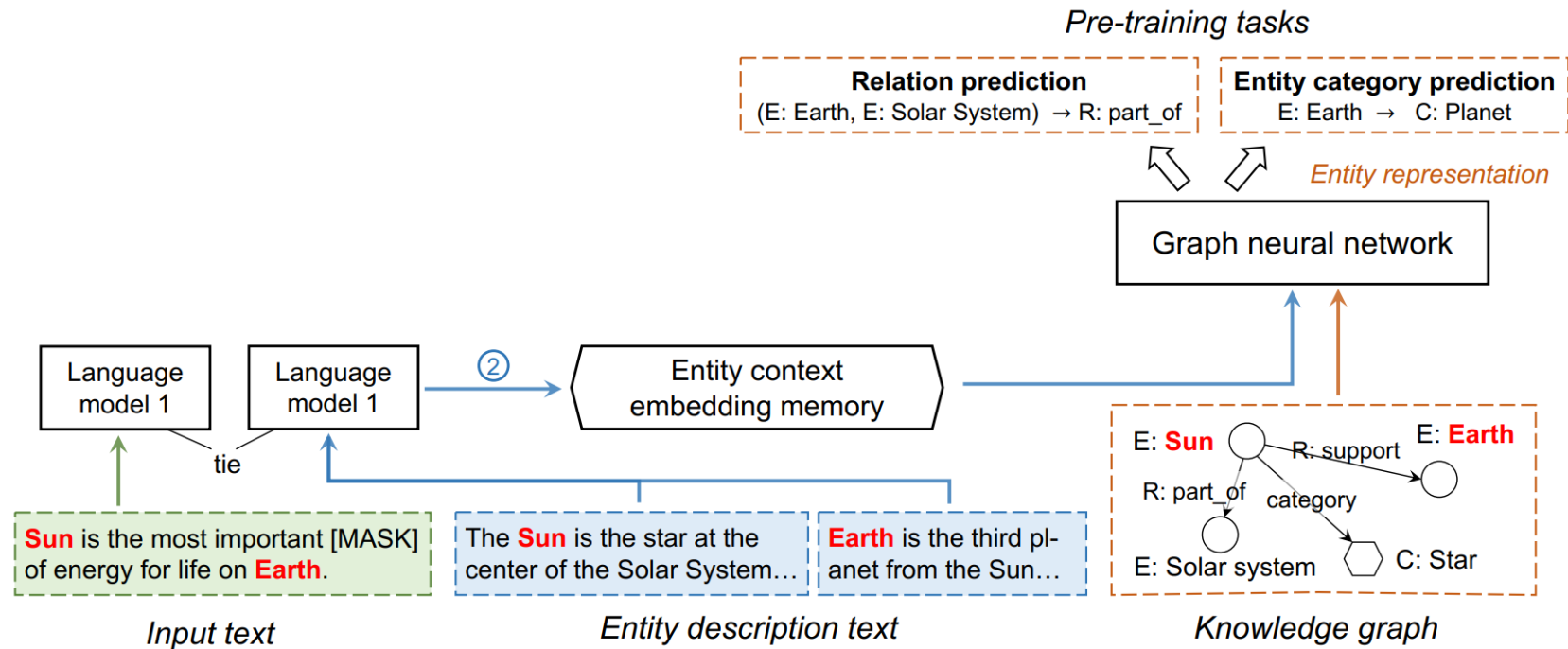
- Goal: capture the knowledge graph structure information
 - TransE based method could not capture the structure information of KG
 - Idea: graph neural network



- [1] D. Yu, C. Zhu, Y. Yang, M. Zeng. 2022. JAKET: Joint Pre-training of Knowledge Graph and Language Understanding (AAAI 22).

JAKET: Joint Pre-training of Knowledge Graph and Language Understanding

□ Goal: capture the knowledge graph structure information

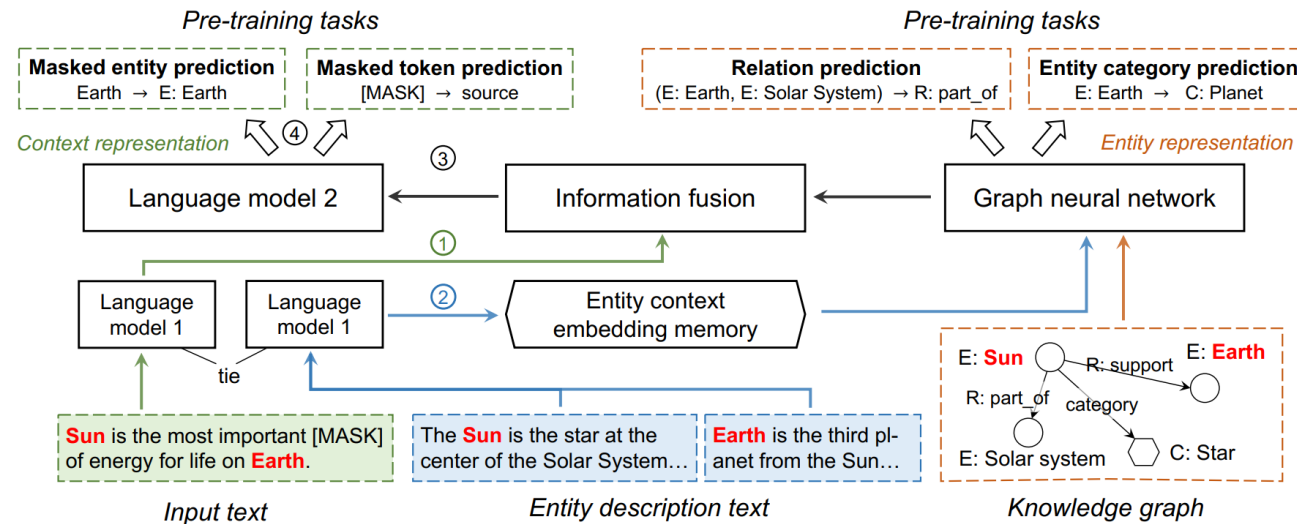


- [1] D. Yu, C. Zhu, Y. Yang, M. Zeng. 2022. JAKET: Joint Pre-training of Knowledge Graph and Language Understanding (AAAI 22).

JAKET: Joint Pre-training of Knowledge Graph and Language Understanding

Advantages:

- Eases the semantic matching between them
- Solves the over-parameterization issue since entity embeddings are no longer part of the model's parameters



- [1] D. Yu, C. Zhu, Y. Yang, M. Zeng. 2022. JAKET: Joint Pre-training of Knowledge Graph and Language Understanding (AAAI 22).

JAKET: Experiments

□ Observations:

- State of the art performance on KE task
- State of the art performance on various NLP tasks

Model	KG-Full		KG-50%	
	1-hop	2-hop	1-hop	2-hop
ERNIE	89.8	70.1	61.2	38.7
KnowBERT	89.5	69.3	61.1	38.3
KEPLER	90.1	70.3	60.7	38.1
CoLAKE	92.4	72.1	62.5	40.6
RoBERTa	90.2	70.8	61.5	39.3
RoB+G+M	91.4	72.6	62.5	40.8
JAKET	93.9	73.2	63.1	41.9

Hits@1 results on the MetaQA dataset.

Model	Training Size		
	100%	20%	5%
GNN	48.2	-	-
RoBERTa	33.4	-	-
RoB+G+M	79.1	66.7	53.5
JAKET	81.6	70.6	58.4

Accuracy results on the entity classification task.

- [1] D. Yu, C. Zhu, Y. Yang, M. Zeng. 2022. JAKET: Joint Pre-training of Knowledge Graph and Language Understanding (AAAI 22).

Roadmap

Part I: Knowledge Graph Reasoning: Basic Concepts and Techniques

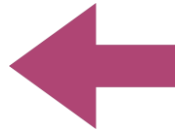
Part II: Recent Advance #1: Neural Reasoning for Natural Language Queries

Part III: Recent Advance #2: Neural Reasoning for Logical Queries

Part IV: Recent Advance #3: Neural Reasoning Beyond Entities and Relations

Part V: Recent Advance #4: LLM+KGR

Part VI: Open Challenges and Future Directions

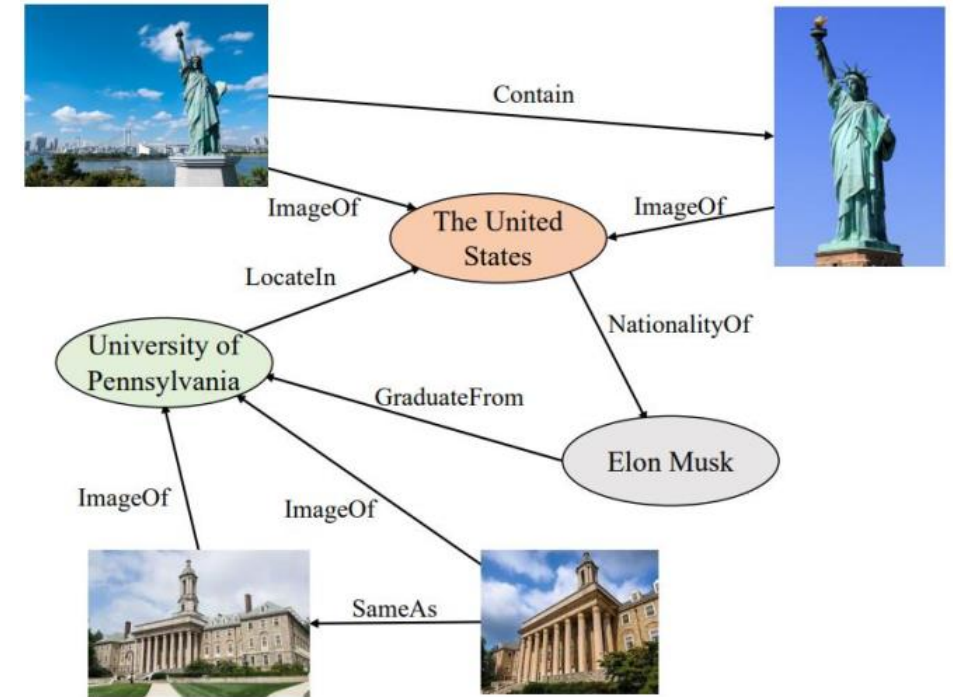


Future Direction #1: Multi-modal knowledge graph reasoning

❑ **Multi-modal knowledge graph reasoning:**
An emerging research area that combines structured knowledge graphs with unstructured data such as images, videos, audio, and other forms of sensory information.

❑ **Possible direction:**

- Knowledge graph reasoning for Visual and Textual Information



Future Direction #2: Retrieval-Augmented Multimodal Language Modeling

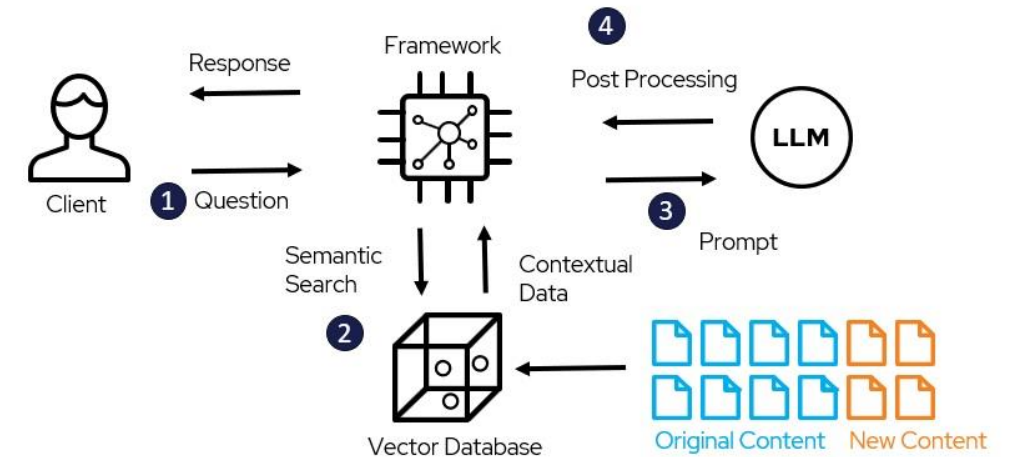
❑ Retrieval augmented generation

- ❑ An AI framework for improving the quality of LLM-generated responses by grounding the model on external sources
- ❑ Mainly focuses on
 - ❑ Text data
 - ❑ Graph data

❑ Possible direction:

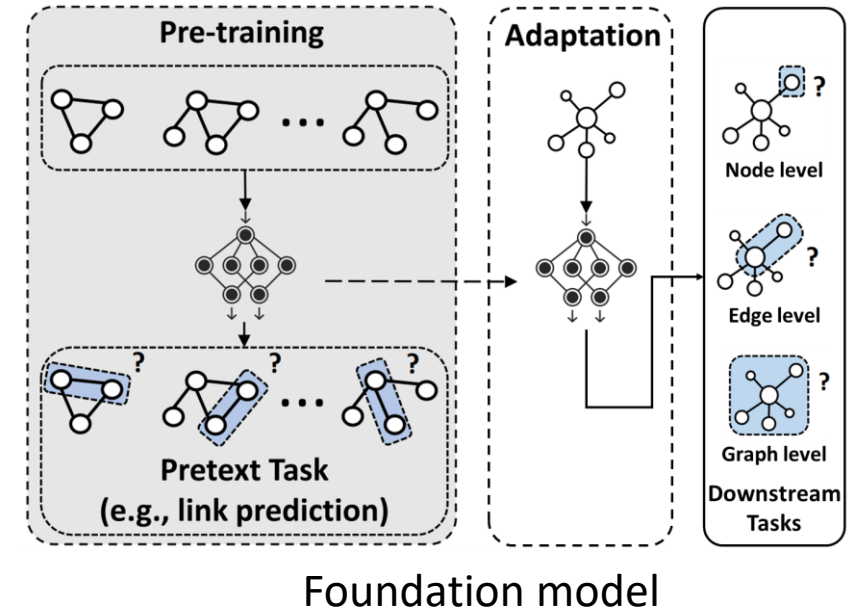
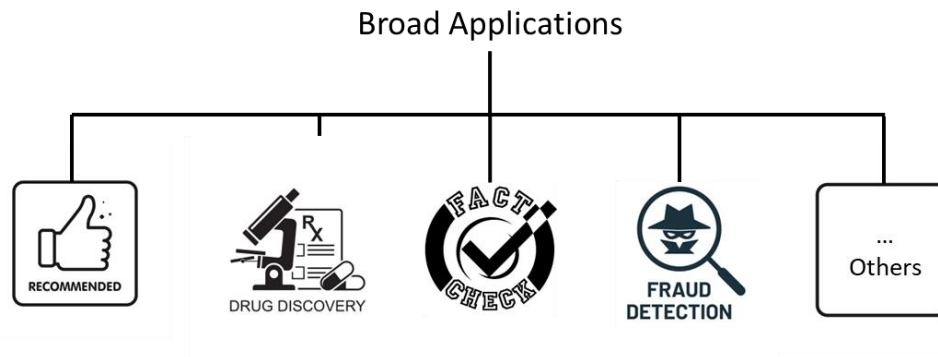
- Integrate other data types
 - Video, audio, image, spatial temporal data...

RAG Architecture Model



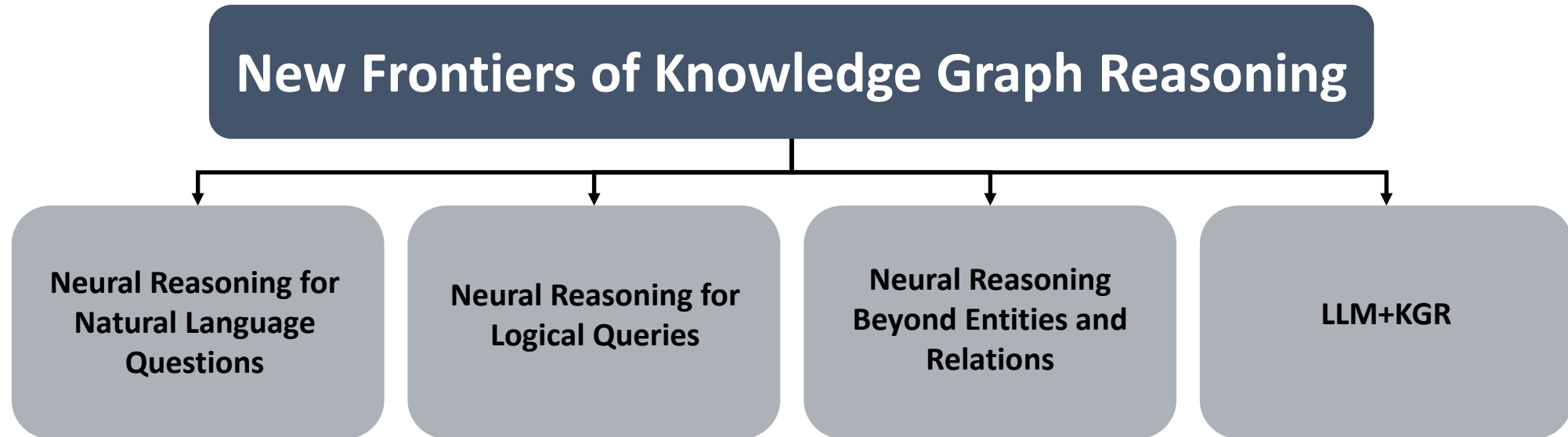
Future Direction #3 Knowledge Graph Foundation Models for Broad Applications

- Knowledge graph reasoning plays a pivotal role in many high-impact applications



- How to build a KG foundation model to solve multiple tasks?

Summarization



Resources

- ❑ **Datasets:** <https://github.com/LIANGKE23/Awesome-Knowledge-Graph-Reasoning#datasets->
- ❑ **Code:** <https://github.com/uma-pi1/kge>
- ❑ **Surveys:**
 - J. Zhang, B. Chen, L. Zhang, X. Ke, H. Ding. Neural, symbolic and neural-symbolic reasoning on knowledge graphs. AI Open, 2021.
 - X Chen, S Jia, Y Xiang. A review: Knowledge reasoning over knowledge graph. Expert Systems with Applications, 2020.
 - K. Liang, L. Meng, M. Liu, Y. Liu, W. Tu, S. Wang, S. Zhou, X. Liu, F. Sun. A Survey of Knowledge Graph Reasoning on Graph Types: Static, Dynamic, and Multimodal. arXiv, 2023.
- ❑ **Related tutorials:**
 - Reasoning on Knowledge Graphs: Symbolic or Neural?
 - <https://aaai2022kgreasoning.github.io/>
 - Rule Induction and Reasoning in Knowledge Graphs
 - <https://www.youtube.com/watch?v=CanRy4NhMGQ>

Acknowledgements

- ❑ Part of the slides are credited to the following authors (in alphabetic order of last name)
 - Endri Kacupaj (Cerence)
 - Magdalena Kaiser (Max-Planck-Institut für Informatik)
 - Jure Leskovec (Stanford University)
 - Apoorv Saxena (Adobe)
 - Komal K. Teru (The Vanguard Group)
- ❑ If you would like to re-use these slides, please contact the original authors

References

- ❑ T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G Bouchard. 2016. Complex Embeddings for Simple Link Prediction (ICML 16).
- ❑ A. Bordes, N. Usunier, and A. Garcia-Duran. 2013. Translating Embeddings for Modeling Multi-relational Data (NeurIPS 13).
- ❑ T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings (AAAI 18).
- ❑ B. Yang, W. Yih, X. He, J. Gao, and L. Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases (ICLR 15).
- ❑ Z. Sun, Z. Deng, J. Nie, and J. Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space (ICLR 19).
- ❑ M. Richardson, and P. Domingos. 2006. Markov Logic Networks (Machine learning 16).
- ❑ W. Cohen. 2016. TensorLog: A Differentiable Deductive Database (arXiv 16).
- ❑ N. Lao, T. Mitchell, and W. Cohen. 2011. Random walk inference and learning in a large scale knowledge base (ACL 11).
- ❑ H. Wang, H. Ren, and J. Leskovec. 2021. Relational message passing for knowledge graph completion (KDD 21).

References

- ❑ W. Xiong, T. Hoang, and WY. Wang. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning (ACL 17).
- ❑ H. Wang, H. Ren, and J. Leskovec. 2021. Relational message passing for knowledge graph completion (KDD 21).
- ❑ F. Yang, Z. Yang, W. Cohen. 2017. Differentiable Learning of Logical Rules for Knowledge Base Reasoning (NeurIPS 17).
- ❑ K. Teru, E. Denis, W.L. Hamilton. 2020. Inductive Relation Prediction by Subgraph Reasoning (ICML 20).
- ❑ B. Antoine, U. Nicolas, G. Alberto, W Jason, and Y. Oksana. 2013. Translating Embeddings for Modeling Multi-relational Data (NIPS 13).
- ❑ S. Hu, L. Zou, J. X. Yu, H. Wang, and D. Zhao. 2018. Answering Natural Language Questions by Subgraph Matching over Knowledge Graphs (TKDE 18).
- ❑ H. Ren, W. Hu and J. Leskovec. 2020. Query2box: Reasoning over Knowledge Graphs in Vector Space using Box Embeddings (ICLR 20).
- ❑ J. Tang. 2022. Neural and Symbolic Logical Reasoning on Knowledge Graphs (AAAI 22).

References

- ❑ L. Liu, B. Du, Y. Fung, H. Ji, J. Xu, and H. Tong. (2021). Kompare: A knowledge graph comparative reasoning system (KDD 21).
- ❑ A. Saxena, A. Tripathi, P. Talukdar. 2020. Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings (ACL 20)
- ❑ L. Liu, B. Du, J. Xu, Y. Xia, H. Tong. 2022. Joint Knowledge Graph Completion and Question Answering (KDD 22).
- ❑ M. Kaiser, R. Saha Roy, and G. Weikum. 2021. Reinforcement Learning from Reformulations in Conversational Question Answering over Knowledge Graphs (SIGIR 21)
- ❑ E. Kacupaj, K. Singh, M. Maleshkova, J. Lehmann. 2022. Contrastive Representation Learning for Conversational Question Answering over Knowledge Graphs (CIKM 22)
- ❑ X. Lv, Y. Gu, X. Han, L. Hou, J. Li, Z. Liu. 2019. Adapting Meta Knowledge Graph Information for Multi-Hop Reasoning over Few-Shot Relations (EMNLP 19).
- ❑ K. Huang, M. Zitnik. 2021. Graph Meta Learning via Local Subgraphs (NeurIPS 21).