# Summary of Knowledge Graph Completion

2024-11-14

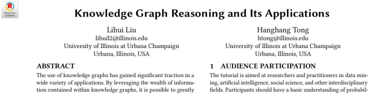
presenters : Sooho Moon & Hunui Lee

DMAIS

# INDEX

# Introduction

- **Before we start...**
  - □ We'ev been studying about **knowledge graph completion(KGC)** since 2024.07.09.
  - □ We would like to extend our gratitude to the authors and our professor whose materials have made our journey both possible and enriching



Knowledge Graph Reasoning and Its Applications

Lihui Liu
lihuil2@illinois.edu
University of Illinois at Urbana Champaign
Urbana, Illinois, USA

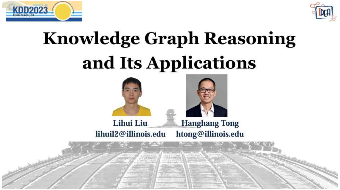
Hanghang Tong
htong@illinois.edu
University of Illinois at Urbana Champaign
Urbana, Illinois, USA

ABSTRACT

The use of knowledge graphs has gained significant traction in a wide variety of applications. By leveraging the wealth of information contained within knowledge graphs, it is possible to greatly

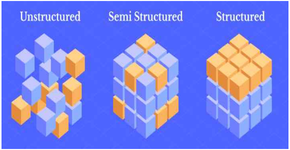
1 AUDIENCE PARTICIPATION

The tutorial is aimed at researchers and practitioners in data mining, artificial intelligence, social science, and other interdisciplinary fields. Participants should have a basic understanding of probabil-

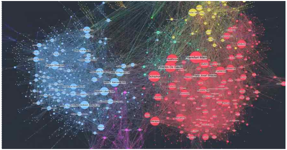


# Introduction

- **What is a KG?**
  - □ First introduced by Google in 2012
  - □ KG, which have long aimed to represent our world through web crawling
  - □ KGs are constructed by using structured, semi structured, unstructured datas



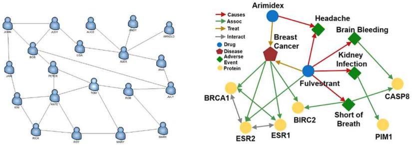

web, JSON, CSV, etc.



NELL, DBpedia, FreeBase, etc.

# Introduction

- **What is a KG?**
  - □ A heterogeneous graph where entities serve as nodes, and edges represent their relationships
  - □ Capable of accommodating **much richer information** than traditional ordinary graphs




△ ordinary graph(left) compared to KG(right)

| Knowledge Graph | Statements | Entities |
|---|---|---|
| YAGO | 120 M | 10 M |
| WIKIDATA | 610 M | 51 M |
| DBpedia | 1.3 B | 6 M |
| GDELT | 3.5 B | 364 M |

△ sizes of popular KGs

# Introduction

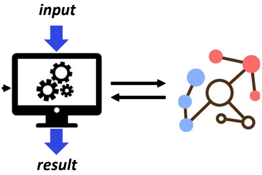
- **Why do we need KGC?**

  □ KGs were mostly incomplete, sparse
  (**60%** of **person** entity did not have **place_of_birth** relation in **DBpedia'14**)

  □ This highlighted the need to fill in gaps to create a more **complete** Knowledge Graph

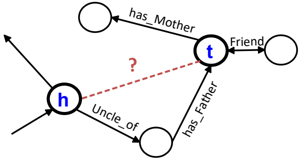
  □ KGC leverages performance in various domains

  ○ information retreival processes in LLMs
  ○ recommender systems
  ○ fact-checking
  ○ question answering

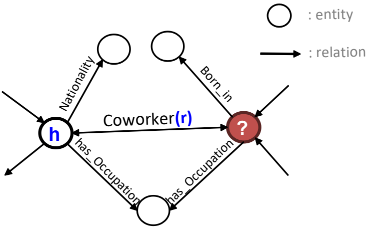


# Introduction

- **Main goal of KGC**
  - □ Given a triplet **(h, r, t)** comprising a head entity **(h)**, a relation **(r)**, and a tail entity **(t)**
  - □ Predict the missing entity or relation to **complete the KG**




△ relation prediction



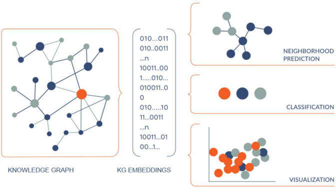

△ link prediction

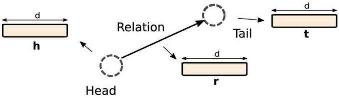
# Introduction

- **General flow of KGC models**
  - □ **Training**
    → Mask the part to be predicted and train the model to rank this masked part as high as possible
  - □ **Testing(**use metrics like AUC, MR, MRR, Hits@k, etc.)
    → Use the trained model to predict the masked part

# Embedding based methods

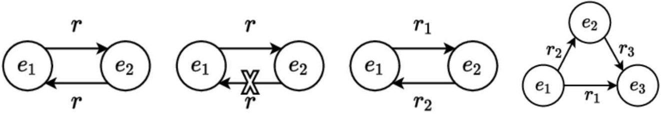
- **What is Embedding based methods?**
  - □ **Goal** : Encode entities and relations as **low-dimensional vectors** in the continuous space
  - □ **Advantages** : Efficient Representation, Ease of representation with vector operation



# Embedding based methods

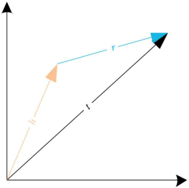
- **Knowledge graph embedding captures KG's patterns**
  - □ Find several relation pattern by embedding entities and relations
  - □ **Symmetry, Antisymmetry, Inversion, Composition** can be captured

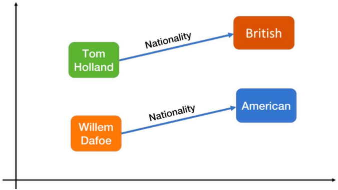



(a) Symmetry (b) Antisymmetry (c) Inversion (d) Composition

# Embedding based methods

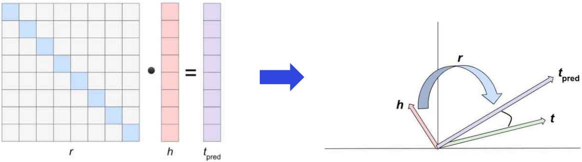
- **TransE (NeurIPS ’13)**
  - □ Make representation with simple operation (light, fast)
  - □ Relation $r$ as a translation from the head entity $h$ to the tail entity $t$ $(t_{pred} = h + r)$



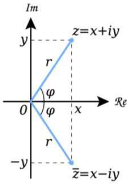
# Embedding based methods

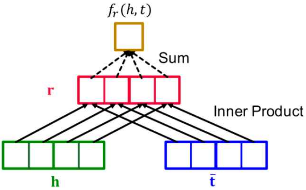
- **DistMult (ICLR '15)**
  - □ Make representation with **dot product**
  - □ Relation $r$ defined as the elementwise weights of the head entity ($h \cdot r = [h_1 \cdot r_1 + \ ... + h_n \cdot r_n]$)



# Embedding based methods

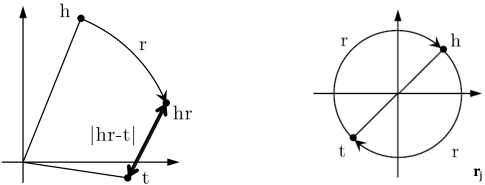
- **ComplEx (ICML '16)**
  - □ Make representation with **Hermitian dot product**
  - □ Using the asymmetry of the Hermitian dot product to represent antisymmetry



# Embedding based methods

- **RotatE (ICLR '19)**
  - ◻ Make representation with **Hadamard product**
  - ◻ Relations modelled as element-wise rotations in complex space ($t_j = h_j r_j, |r_j| = 1$)




$\mathbf{r}_{\mathrm{j}} = -1$ or $\theta_{r,j} = \pi$

# Embedding based methods

- **The patterns a model can capture depends on the representation method**

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

# Embedding based methods(overview)

| TransE | DistMult | ComplEx | RotatE |
|---|---|---|---|
| 2013 | 2015 | 2016 | 2019 |

## ▪ Advantages of KGE

☐ Complex structures can be represented by vector operation

☐ Pattern Learning

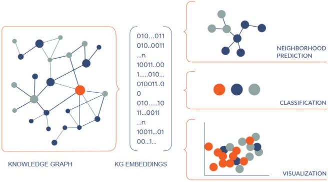
☐ Effect representation

## ▪ Potential bottleneck in KGE

☐ Entity-specific work

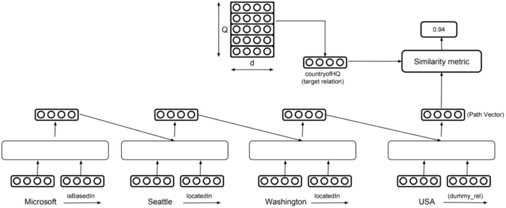
☐ Not applicable in inductive setting

☐ No structural information used



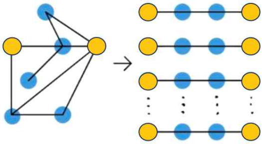
# Path based methods

- **What is Path based methods?**
  - ☐ **Goal** : reach a target entity and inferring new relationships by **exploring multiple paths** in the KG
  - ☐ **Advantages** : inferring without explicit rules



# Path based methods

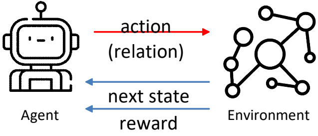
- **PRA (ACL ’11)**
  - ☐ Deriving multiple possible paths between entity pairs through **random walks**
  - ☐ Use supervised training to rank different paths

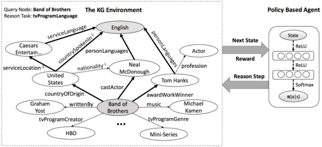


| | Path 1 | Path 2 | ... | Path n | Label |
|---|---|---|---|---|---|
| Query 1 | Score 1.1 | Score 1.2 | ... | Score 1.n | y1 |
| Query 2 | ... | ... | ... | ... | y2 |
| ... | ... | ... | ... | ... | ... |
| Query k | Score k.1 | Score k.2 | ... | Score k.n | yk |

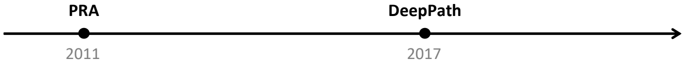
# Path based methods

- **DeepPath (ACL '17)**
  - ☐ Exploring path in KG using **reinforcement learning agent**, modeled as a Markov Decision Process
  - ☐ Agent learns optimal paths to target entities by following reward to discovering efficient paths
  - ☐ Learned paths can be represented as logical rules and used for inferring

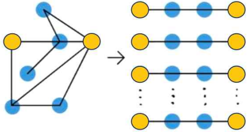


# Path based methods(overview)




- **Advantages of Path based methods**
  - ☐ Generalization and inference with indirect connection between entities
  - ☐ Path can be interpreted as logical rule



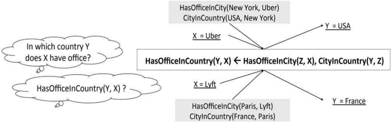
- **Potential bottleneck in Path based**
  - ☐ Only rely on observed paths
  - ☐ Scarcely utilize graph structural informations

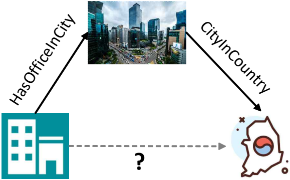
# Rule mining methods

## ▪ What is rule mining?

☐ **Goal** : Aiming to extract **meaningful first order rules** that can be applied to new, unseen data

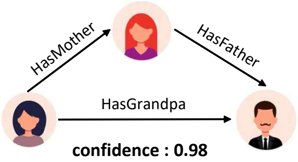
☐ **Advantages** : Generalizable, explainable

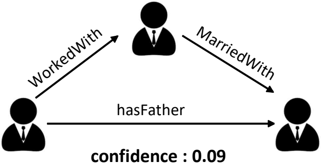


# Rule mining methods

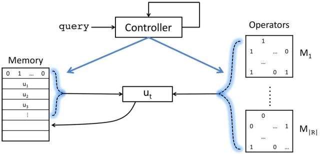
- **Rule mining extracts two components for reasoning**
  - ☐ **The rule itself**(HasOfficeInCountry(Y, X) ⇐ HasOfficeInCity(Z, X), CityInCountry(Y, Z))
  - ☐ **Confidence of individual rule**(how much can we trust it?)



# Rule mining methods

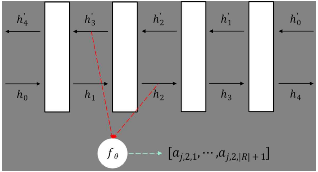
- **NeuralLP(NeurIPS '17)**
  - ☐ **First end-to-end differentiable approach** to learning logical rules
  - ☐ A **LSTM** system that mine rules with varying lengths




$$
\begin{aligned}
\mathbf{h_t} &= \text{update}\left(\mathbf{h_{t-1}}, \text{input}\right) \\
\mathbf{a_t} &= \text{softmax}\left(W\mathbf{h_t} + b\right) \\
\mathbf{b_t} &= \text{softmax}\left(\left[\mathbf{h_0}, \ldots, \mathbf{h_{t-1}}\right]^T \mathbf{h_t}\right)
\end{aligned}
$$

# Rule mining methods

- **DRUM(NeurIPS '19)**
  - ☐ Highlights that NeuralLP can mine **incorrect rules with high confidence**
  - ☐ A **bidirectional RNN** system to reduce the bottleneck of NeuralLP

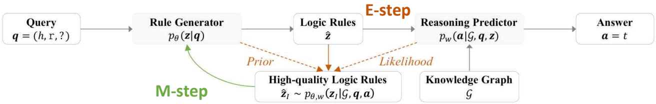



$$\mathbf{h}_i^{(j)}, \mathbf{h}'^{(j)}_{T-i+1} = \mathbf{BiRNN}_j(\mathbf{e}_H, \mathbf{h}_{i-1}^{(j)}, \mathbf{h}'^{(j)}_{T-i}),$$

$$[a_{j,i,1}, \cdots, a_{j,i,|\mathcal{R}|+1}] = f_\theta([\mathbf{h}_i^{(j)}, \mathbf{h}'^{(j)}_{T-i+1}]),$$

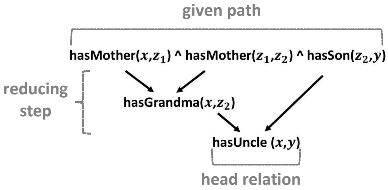
# Rule mining methods

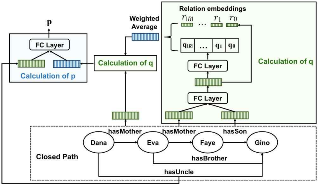
- **RNNLogic(ICLR '21)**
  - ☐ Points out the problem of **large action space** in previous models
  - ☐ An **EM algorithm-based optimization** rule mining model
  - ☐ Seperates rule generating from rule reasoning



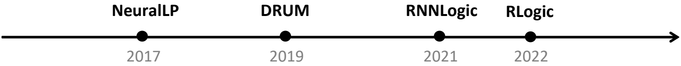
# Rule mining methods

- **RLogic(KDD '22)**
  - ☐ Identifies the issue that **previous models can't mine unseen rules**
  - ☐ Proposes to embrace the deductive reasoning for rule mining
  - ☐ A recursive framework that reduces a path to a single head relation



# Rule mining methods(overview)

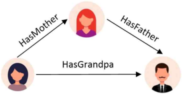



## ▪ Advantages of rule mining

☐ Mined rules can be generalized

☐ Rules are explainable and understandable

☐ Entity inductive framework




**VS**

[0.013472648337483406, -0.005545202642679214, 0.04156069457530975,
-0.01467960886657238, -0.007239829283207655, -0.008201991207897663,
0.009404981508851051, -0.02347548119723797, -0.005465532653033733,
-0.006886753719300032, 0.04091925546526909, 0.010529556311666965,
0.017471831291913986, 0.005145607516169548, 0.007516753394156694,
0.004749914165586233, -0.04658839479088783, -0.0037852816749364138,
0.03419521823525429, 0.001870729262009263, 0.03879108652472496,
-0.02726254053413868, 0.0177603866904974, -0.0429973378777504,
0.009313264861702919, 0.016721446067094803, -0.04639590531587601, . . . ]

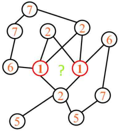
## ▪ Potential bottleneck in rule mining

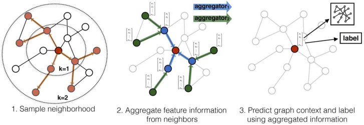
☐ Rules are inherently discrete

☐ The approach does not align well with the incompleteness of KG

# GNN based methods

- **What is GNN based methods?**
  - ☐ **Goal** : Integrates GNN approaches(SEAL, GraphSAGE, etc.) into KGC
  - ☐ **Advantages** : Leverages structural information of graph



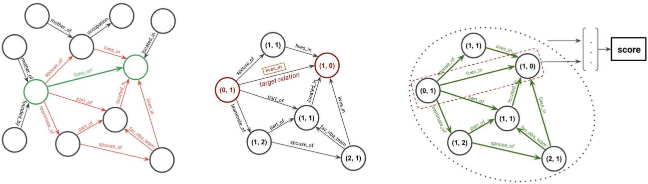




△ structural node labelling(SEAL)

△ aggregating process(GraphSAGE)

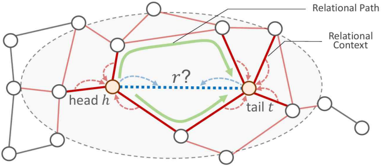
# GNN based methods

- **GraIL(ICML '20)**
  - ☐ Adopts subgraph reasoning around target nodes to enhance relational understanding
  - ☐ Performs GNN message passing with structurally labeled nodes to infer missing relations

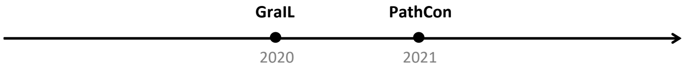


# GNN based methods

- **PathCon(KDD '21)**
  - ☐ Acknowledges that reasoning can be decomposed into **context** and **path**
  - ☐ **Context** : defines the entity type of target nodes through relatoinal message passing
  - ☐ **Path** : defines the reletive position between target nodes



# GNN based methods(overview)

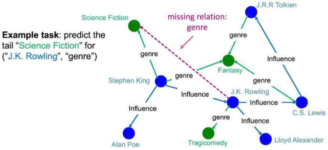


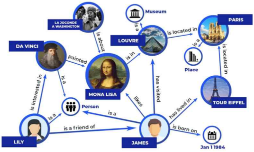

- **Advantages of GNN based methods**
  - ☐ Combines successful methods previously introduced for ordinary graphs
  - ☐ Entity inductive framwork

- **Potential bottleneck in GNN based methods**
  - ☐ Challenging since edge type also needs to be modeled

# Conclusion

**We view** knowledge graph completion not merely as a method for filling gaps in the knowledge graph, but as a tool **enabling AI models to understand** relationships between real-world entities, ultimately fostering a deeper understanding of the world we live in



# Thank You!