REPRESENTATION LEARNING ON KNOWLEDGE GRAPHS: FROM SHALLOW EMBEDDING TO GRAPH NEURAL NETWORKS

Yizhou Sun
Department of Computer Science
University of California, Los Angeles
yzsun@cs.ucla.edu

November 8, 2020

Roadmap

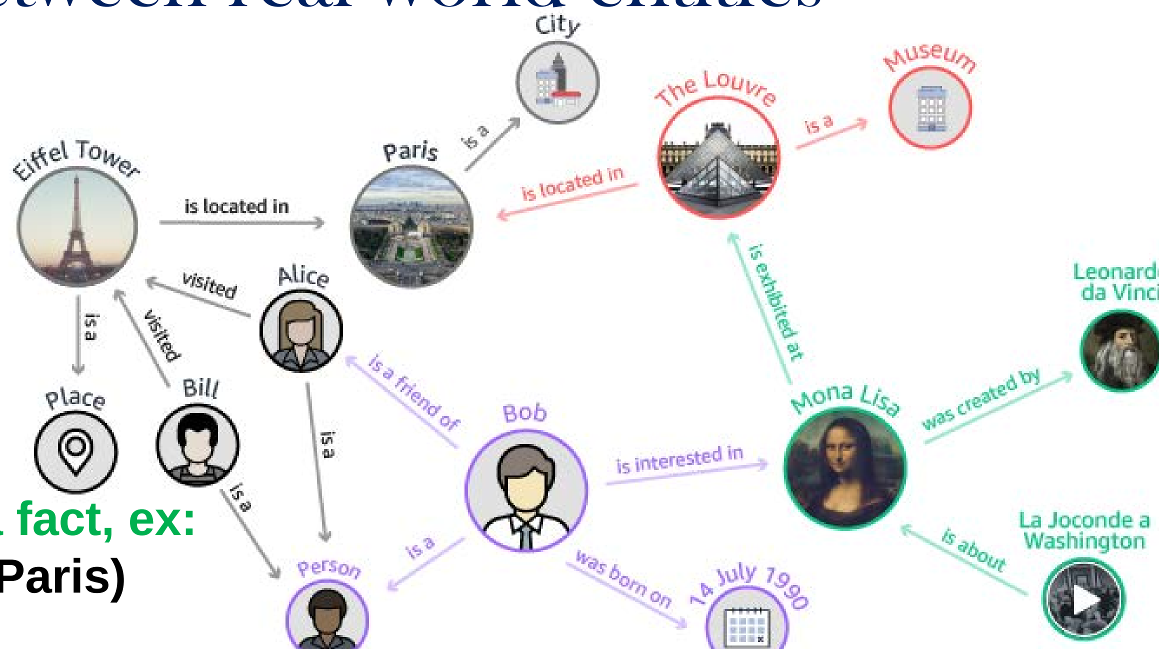
- PARI I:
  - Shallow knowledge graph embedding
- PART II:
  - Bringing additional symbolic knowledge into knowledge graph embedding
- PART III:
  - Graph neural networks

PART I

- Shallow knowledge graph embedding

Knowledge Graph

- What are knowledge graphs?
  - Multi-relational graph data
    - (heterogeneous information network)
  - Provide structured representation for semantic relationships between real-world entities



A triple (h, r, t) represents a fact, ex:
(Eiffel Tower, is located in, Paris)

Examples of KG

General-purpose KGs

Bio & Medical KGs

Product Graphs & E-commerce

Common-sense KGs & NLP

Applications of KGs

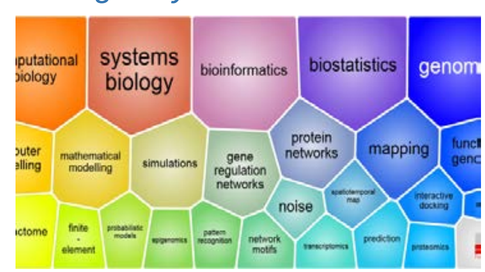
- Foundational to knowledge-driven AI systems
- Enable many downstream applications (NLP tasks, QA systems, etc)

Natural Language Processing

QA & Dialogue systems

Knowledge Graphs

Recommendation Systems



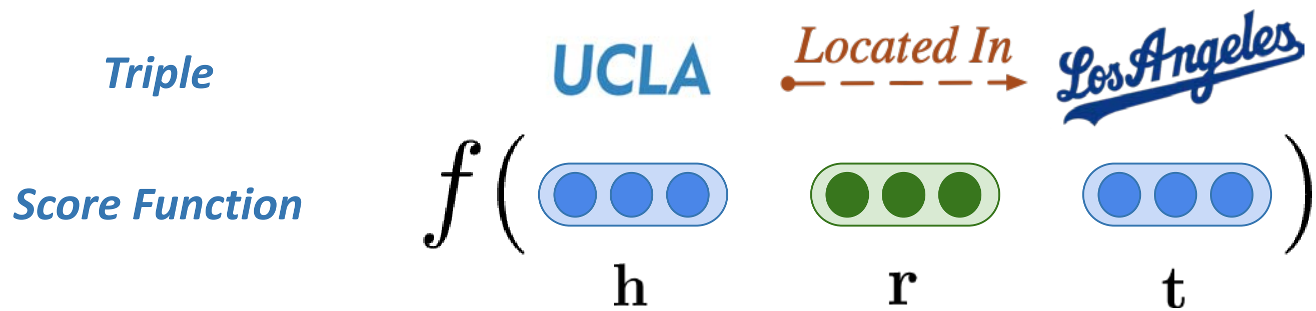
Computational Biology

Knowledge Graph Embedding

- Goal:
  - Encode entities as low-dimensional vectors and relations as parametric algebraic operations
- Applications:
  - Dialogue agents
  - Question answering
  - Machine comprehension
  - Recommender systems
  - …

Key Idea of KG embedding algorithms

- Define a score function for a triple: $f_r(\boldsymbol{h}, \boldsymbol{t})$
  - According to entity and relation representation



- Define a loss function to guide the training
  - E.g., an observed triple scores higher than a negative one

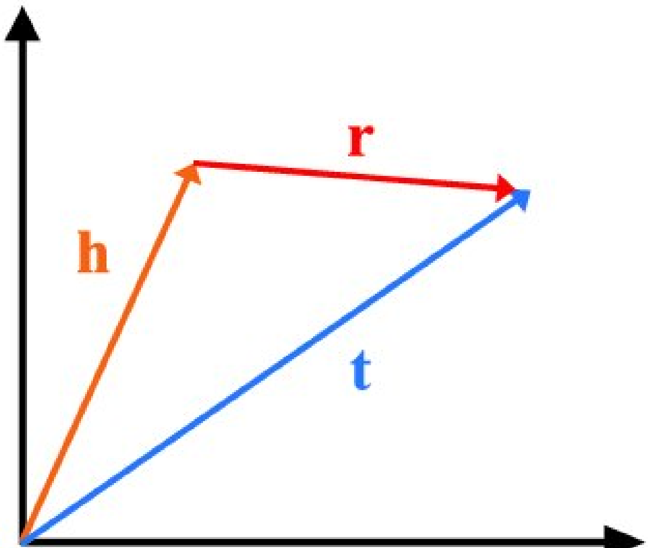
Summary of Existing Approaches

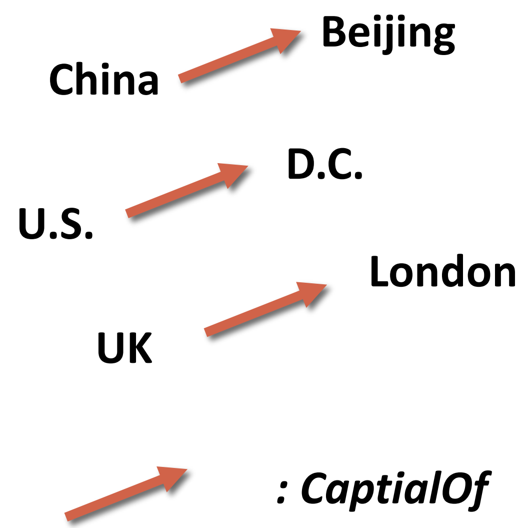
Model	Score Function	
SE (Bordes et al., 2011)	$-\|\boldsymbol{W}_{r,1}\mathbf{h} - \boldsymbol{W}_{r,2}\mathbf{t}\|$	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^k, \boldsymbol{W}_{r,\cdot} \in \mathbb{R}^{k \times k}$
TransE (Bordes et al., 2013)	$-\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
TransX	$-\|g_{r,1}(\mathbf{h}) + \mathbf{r} - g_{r,2}(\mathbf{t})\|$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
DistMult (Yang et al., 2014)	$\langle \mathbf{r}, \mathbf{h}, \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
ComplEx (Trouillon et al., 2016)	$\mathrm{Re}(\langle \mathbf{r}, \mathbf{h}, \overline{\mathbf{t}} \rangle)$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k$
HolE (Nickel et al., 2016)	$\langle \mathbf{r}, \mathbf{h} \otimes \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
ConvE (Dettmers et al., 2017)	$\langle \sigma(\mathrm{vec}(\sigma([\overline{\mathbf{r}}, \overline{\mathbf{h}}] * \boldsymbol{\Omega}))\boldsymbol{W}), \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
RotatE	$-\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|^{\boxed{2}}$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k, \|r_i\| = 1$

Source: Sun et al., RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space (ICLR'19)

TransE: Score Function

- Relation: translating embedding





- Score function
  - $f_r(\boldsymbol{h}, \boldsymbol{t}) = -\left|\left|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\right|\right| = -d(\boldsymbol{h} + \boldsymbol{r}, \boldsymbol{t})$

Bordes et al., Translating embeddings for modeling multi-relational data, NeurIPS 2013

TransE: Objective Function

- Objective Function
  - Margin-based ranking loss
  - $L = \sum_{(h,r,t) \in S} \sum_{(h',r,t') \in S'_{(h,r,t)}} [\gamma + d\,(\boldsymbol{h} + \boldsymbol{r}, \boldsymbol{t}) - d(\boldsymbol{h}' + \boldsymbol{r}, \boldsymbol{t}')]_+$
    - $[x]_+$ denotes the positive part of x, i.e., $\max(0, x)$
    - $\gamma > 0$ denotes the margin hyperparameter
      - The higher the bigger difference between positive triple and negative one
    - S: positive triple set; S': corrupted triple set (negative triples)
- Optimization: stochastic gradient descent

TransE: Limitations

- One-one mapping: $t = \phi_r(h)$
  - Given (h,r), t is unique
  - Given (r,t), h is unique
- Anti-symmetric
  - If r(h,t) then r(t,h) is not true
  - Cannot model symmetric relation, e.g., friendship
- Anti-reflexive
  - r(h,h) is not true
  - Cannot model reflexive relations, e.g., synonym

DistMult

- Bilinear score function
  - $f_r(\boldsymbol{h}, \boldsymbol{t}) = \boldsymbol{h}^T \boldsymbol{M}_r \boldsymbol{t}$
    - Where $\boldsymbol{M}_r$ is a diagonal matrix with diagonal vector $\boldsymbol{r}$
  - A simplification to neural tensor network (NTN)
- Objective function
  - $L = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'_{(h,r,t)}} [\gamma - f_r(\boldsymbol{h}, \boldsymbol{t}) + f_r(\boldsymbol{h}', \boldsymbol{t}')]_+$
- Limitation
  - Can only model symmetric relation
    - $f_r(\boldsymbol{h}, \boldsymbol{t}) = f_r(\boldsymbol{t}, \boldsymbol{h})$

Yang et al., Embedding entities and relations for learning and inference in knowledge bases, ICLR 2015

PART II

- Bringing additional symbolic knowledge into knowledge graph embedding

Outline

- Introduction
- Bringing First-Order Logic into Uncertain KG Embedding
- Bringing Ontological Concepts and Meta Relations into KG Embedding
- Summary & Future Work

Limitations

- Closed-world assumption
  - Observed triples are true facts
  - Unseen triples are false
- Flat structure assumption
  - No additional structures
  - Every triple is scored using the same form of score function

Solutions

- From deterministic KGs to uncertain KGs
  - Bringing logic rules and probabilistic soft logic to handle uncertainty
  - Examples of uncertain KGs

NELL

[Mitchell et al. 2018]

ConceptNet

An open, multilingual knowledge graph

- From one-view KGs to two-view KGs
  - Bringing ontological concepts and meta relations

Outline

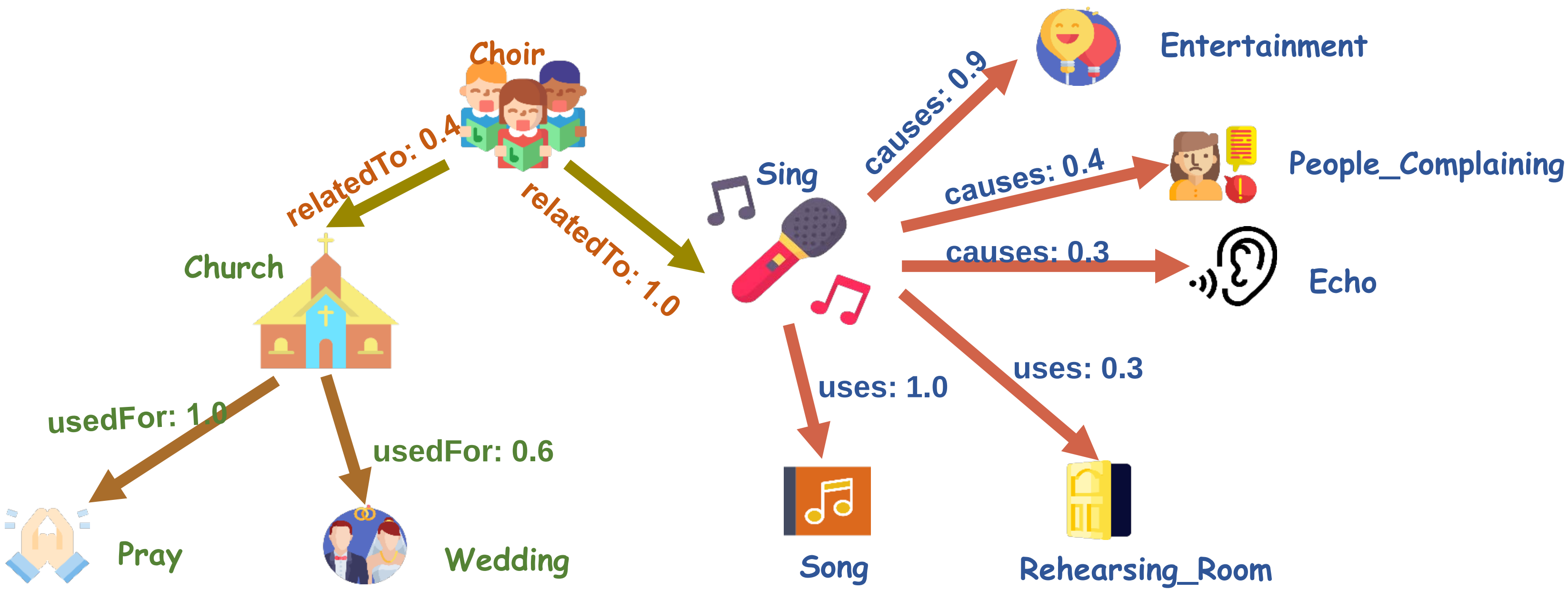
- Introduction
- Bringing First-Order Logic into Uncertain KG Embedding
  - Chen et al., "*Embedding Uncertain Knowledge Graphs*," AAAI'19
- Bringing Ontological Concepts and Meta Relations into KG Embedding
- Summary & Future Work

Two Types of Errors in KG

- False positive
  - An observed triple is wrong,
    - e.g., (Obama, is_born_in, Kenya)
- False negative
  - A true fact is missing
    - e.g., (Eiffel Tower, is located in, France)

Handling Uncertainty in Triples

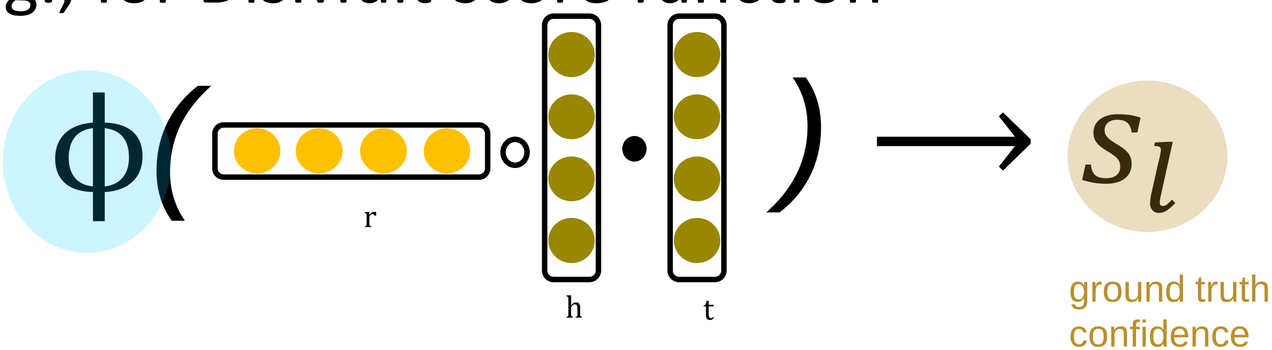
- False positive errors can be alleviated by introducing uncertainty
  - E.g., (Obama, is_born_in, Kenya): 0.01



  - Fit $f_r(\boldsymbol{h}, \boldsymbol{t})$ to uncertainty scores

From score function to uncertainty score

- Given a triple $l = (h, r, t)$ with uncertainty score s_l
  - Transform $f_r(\boldsymbol{h}, \boldsymbol{t})$ into a score in the range [0,1]
    - E.g., for DisMult score function

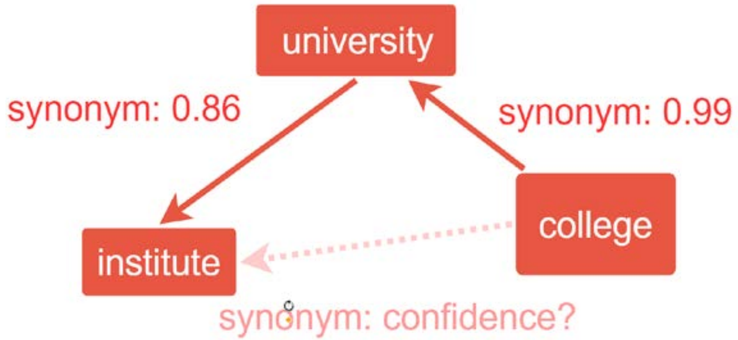


    - Where $\phi(\cdot)$ can be defined as

- **Logistic function** $\phi(x) = \dfrac{1}{1 + e^{-(\mathbf{w}x + \mathbf{b})}}$ **UKGE(logi)**
- **Bounded Rectifier** $\phi(x) = \min(\max(\mathbf{w}x + \mathbf{b}, 0), 1)$ **UKGE(rect)**

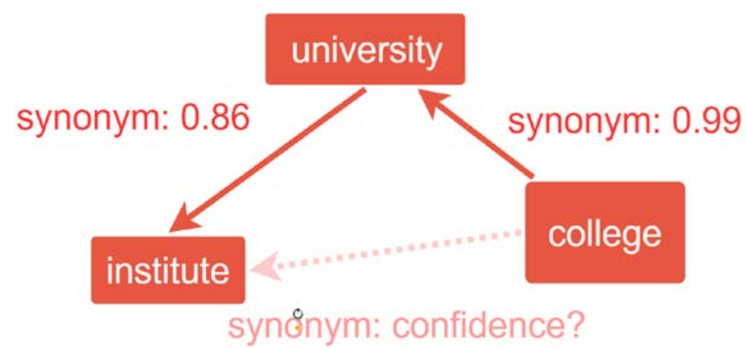
Handling Missing Facts

- Are unseen triples still needed?
  - Yes, negative triples are still data points!
- Can we treat them as false, i.e., $s_l = 0$, if triple l is unseen?
  - No, we are going to make too many mistakes!
    - The potential probability of an unseen triple could be higher than an observed triple with low confidence



Bringing Logic Rules

- What are logic rules?
  - Logic rule
    - (A , synonym, B) ∧ (B , synonym, C) → (A, synonym, C)
  - Ground rule
    - (college, synonym, university) ∧ (university , synonym, institute) → (college, synonym, institute)
- Why are they helpful?
  - Help us to infer the score for unseen triples



Probabilistic Soft Logic

- Quantify a ground rule using PSL
  - Lukasiewicz t-norm, from Boolean logic to soft logic

$$l_1 \wedge l_2 = \max\{0, I(l_1) + I(l_2) - 1\}$$
$$l_1 \vee l_2 = \min\{1, I(l_1) + I(l_2)\}$$
$$\neg l_1 = 1 - I(l_1)$$

  - Probability of a ground rule $\gamma \equiv \gamma_{body} \rightarrow \gamma_{head}$
    - $p_\gamma = I(\neg\gamma_{body} \vee \gamma_{head}) = \min\{1, 1 - I(\gamma_{body}) + I(\gamma_{head}))$
  - Distance to satisfaction
    - $d_\gamma = 1 - p_\gamma = \max\{0, I(\gamma_{body}) - I(\gamma_{head})\}$

More publications on PSL: https://psl.linqs.org/

The Goal: Minimize Distance to Satisfaction

- Example: Consider the following ground rule

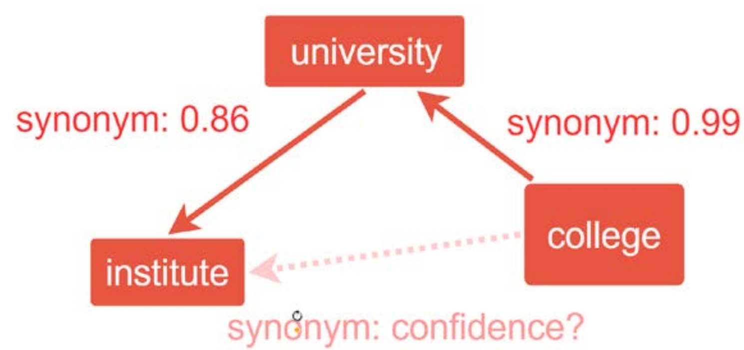
  l_1 confidence: 0.99 $\qquad$ l_2 confidence: 0.86

  - (college, synonym, university) ∧ (university , synonym, institute) → (college, synonym, institute)

  l_3 confidence: ?

  - Recall, $d_\gamma = 1 - p_\gamma = \max\{0, I(\gamma_{body}) - I(\gamma_{head})\}$

$$d_\gamma = \max\{0, \boxed{I(l_1 \wedge l_2)} - I(l_3)\}$$
$$= \max\{0, \overset{0.99}{s_{l_1}} + \overset{0.86}{s_{l_2}} - 1 - f(l_3)\}$$
$$= \max\{0, 0.85 - f(l_3)\}$$



Say, our embedding model predicts it as 0.65.
How good is this prediction?

The New Embedding Model

- For observed triples, force its score close to ground truth score
- For unseen triples, minimize the distance to satisfaction in ground rules they are involved

$$\mathcal{J} = \sum_{l \in \mathcal{L}^+} |\boxed{f(l)} - s_l|^2 + \sum_{l \in \mathcal{L}^-} \sum_{\gamma \in \Gamma_l} |\boxed{\psi_\gamma(f(l))}|^2$$

Embedding-based confidence function

Distance to satisfaction for a ground rule γ, where $triple\ l$ is involved in

Experiments

- Datasets

Dataset	#Ent.	#Rel.	#Rel. Facts	Avg(s)	Std(s)
CN15k	15,000	36	241,158	0.629	0.232
NL27k	27,221	404	175,412	0.797	0.242
PPI5k	5,000	7	271,666	0.415	0.213

- Logic Rules

(A, relatedto, B)∧(B,relatedto,C)→(A,relatedto,C)
(A, causes, B)∧(B,causes,C)→(A,causes,C)

(A, competeswith, B)∧(B,competeswith,C)→(A,competeswith,C)
(A, atheletePlaysForTeam, B)∧(B, teamPlaysSports, C)→(A, atheletePlaysSports, C)

(A, binding, B)∧(B,binding,C)→(A,binding,C)

Baselines

- Deterministic KG embedding models, which does not model confidence scores explicitly
  - TransE [Bordes et al. 2013)]
  - DistMult [Yang et al. 2015]
  - ComplEx [Trouillon et al. 2016]
- Uncertain Graph Embedding, which only provides node embeddings
  - URGE [Hu et al. 2017]
- Two simplified version of our models
  - Without Negative Sampling (UKGE_n-)
    - Can we just ignore the negative links during training?
  - Without PSL (UKGE_p-)
    - Will simply treating unseen relations as 0 a good strategy?

Relation Fact Confidence Score Prediction

- Given an unseen triple (h,r,t), predict its confidence
- Metrics: MSE and MAE ($\times 10^{-2}$)

Dataset	CN15k		NL27k		PPI5k	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE
URGE	10.32	22.72	7.48	11.35	1.44	6.00
UKGE_{n-}	23.96	30.38	24.86	36.67	7.46	19.32
UKGE_{p-}	9.02	20.05	2.67	7.03	0.96	4.09
UKGE_{rect}	**8.61**	**19.90**	**2.36**	**6.90**	**0.95**	**3.79**
UKGE_{logi}	9.86	20.74	3.43	7.93	0.96	4.07

Relation Fact Ranking

- Given a query (h, r, ?t), rank all entities in our vocabulary as tail candidates
- Metrics: normalized Discounted Cumulative Gain (nDCG) (linear gain and exp gain)

metrics	CN15K		NL27k		PPI5k	
Dataset	linear	exp.	linear	exp.	linear	exp.
TransE	0.601	0.591	0.730	0.722	0.710	0.700
DistMult	0.689	0.677	0.911	0.897	0.894	0.880
ComplEx	0.723	0.712	0.921	0.913	0.896	0.881
URGE	0.572	0.570	0.593	0.593	0.726	0.723
UKGE_{n-}	0.236	0.232	0.245	0.245	0.514	0.517
UKGE_{p-}	0.769	0.768	0.933	0.929	0.940	0.944
UKGE_{rect}	0.773	0.775	0.939	0.942	0.946	0.946
UKGE_{logi}	**0.789**	**0.788**	**0.955**	**0.956**	**0.970**	**0.969**

Relation Fact Ranking – Case Study

Dataset	Head	Relation	Ground Truth: Entity	Ground Truth: Score	Predictions: Entity	Predictions: Predicted Score	Predictions: True Score
CN15k	house	usedfor	sleeping	1.0	**relaxing**	**0.86**	**N/A**
			rest	0.98	sleeping	0.85	1.0
			bed away from home	0.71	rest	0.82	0.98
			stay overnight	0.71	**hotel room**	**0.80**	**N/A**
NL27k	Toyota	competeswith	Honda	1.0	Honda	0.94	1.0
			Ford	1.0	Hyundai	0.91	0.72
			BMW	0.96	**Chrysler**	**0.90**	**N/A**
			General Motors	0.90	Nissan	0.89	0.86

Outline

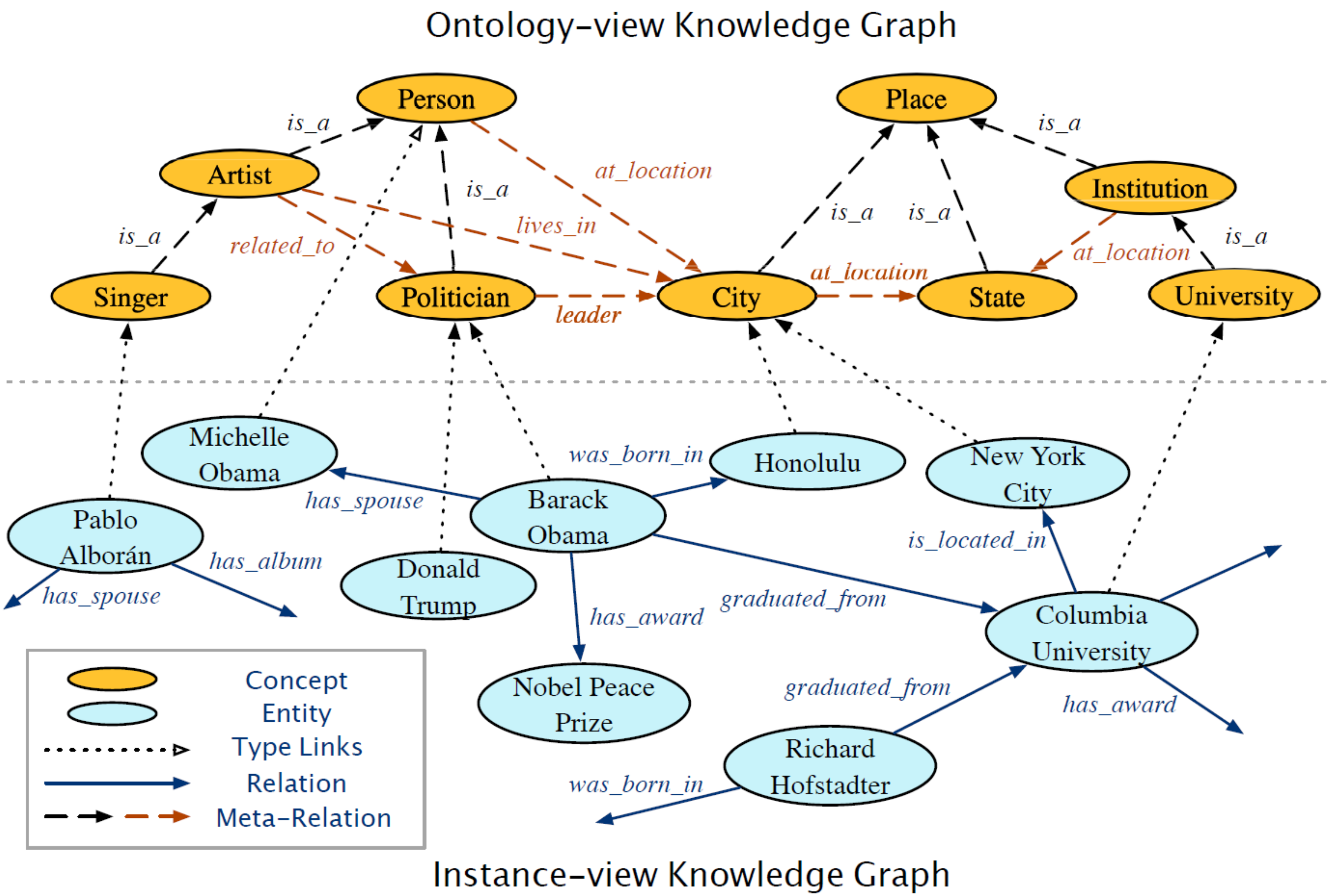
- Introduction
- Bringing First-Order Logic into Uncertain KG Embedding
- Bringing Ontological Concepts and Meta Relations into KG Embedding
  - Hao et al., "*Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts*," KDD'19
- Summary & Future Work

Why ontological view?

- Meta-Level reasoning
  - What kind of relations would a scientist has?
    - Works in universities or research labs
    - Graduated from some university
    - …

- Bring more information to instances, which especially benefits long-tail entities
  - E.g., given Anna is a scientist, she should be close to other scientists in the embedding space

Instance View and Ontological View of KG

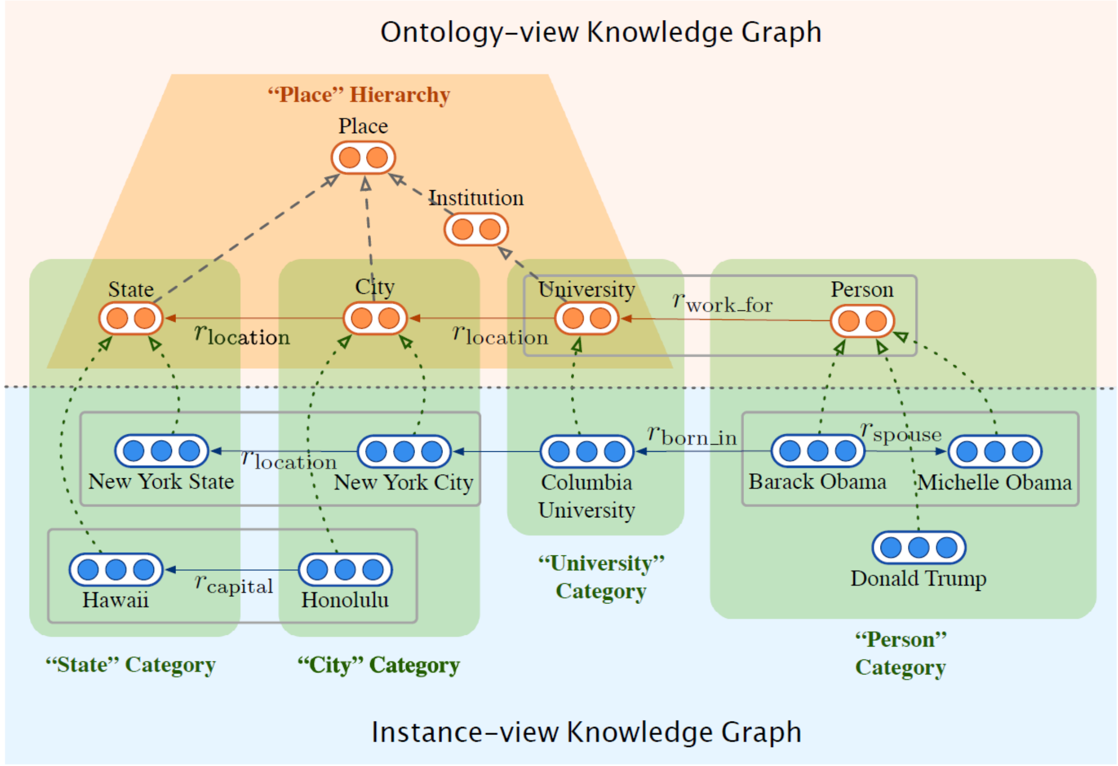


More on Ontological View

- Relation to Schema
  - Schema provides a template or guidance on what types of relation could hold for a specific pair of entity types
  - Also potentially with hierarchical taxonomy
- How to get it?
  - Integrate KG with other sources
    - E.g., align YAGO with ConceptNet

Joint Embedding

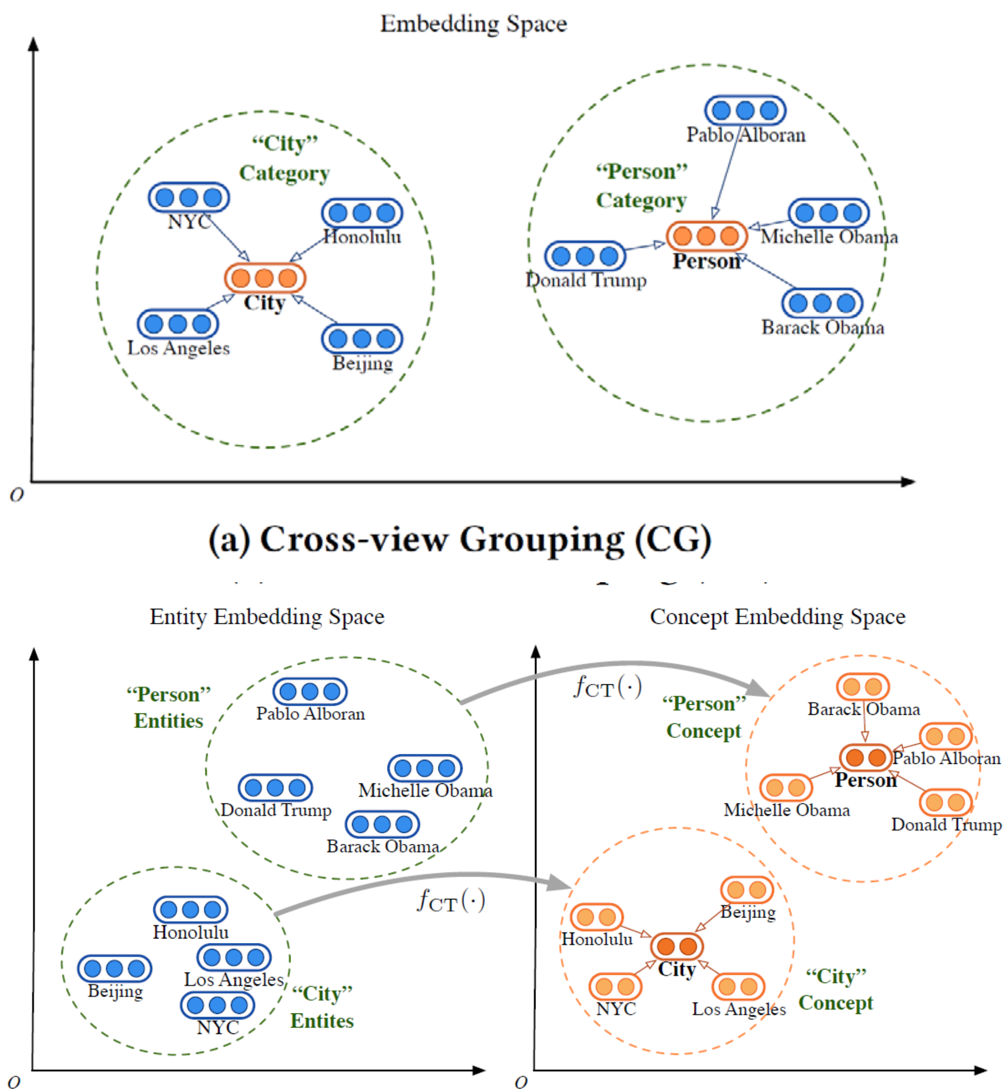
- Instance embedding provide detailed and rich information for their corresponding ontological concepts
- Ontological concepts largely determine the embedding of their instances



Cross-View Association Model

- Based on cross-view links, associate the instance embedding space and ontological embedding space
  - Option 1 (**Cross-View Grouping, CG**): force the two spaces into the same space
    - $$J_{\text{Cross}}^{\text{CG}} = \frac{1}{|\mathcal{S}|} \sum_{(e,c)\in\mathcal{S}} \left[||\mathbf{c} - \mathbf{e}||_2 - \gamma^{\text{CG}} \right]_+$$
  - Option 2 (**Cross-View Transformation, CT**): transform instance space into ontological space
    - $$J_{\text{Cross}}^{\text{CT}} = \frac{1}{|\mathcal{S}|} \sum_{\substack{(e,c)\in\mathcal{S} \\ \wedge(e,c')\notin\mathcal{S}}} \left[\gamma^{\text{CT}} + ||\mathbf{c} - f_{\text{CT}}(\mathbf{e})||_2 - \left\|\mathbf{c}' - f_{\text{CT}}(\mathbf{e})\right\|_2 \right]_+$$

Illustration of CG and CT



(a) Cross-view Grouping (CG)

(b) Cross-view Transformation (CT)

Hierarchy-Aware Intra-View Model

- Base models could be any existing KG embedding models
  - Examples:

$$f_{\mathrm{TransE}}(\mathbf{h}, \mathbf{r}, \mathbf{t}) = -||\mathbf{h} + \mathbf{r} - \mathbf{t}||_2$$

$$f_{\mathrm{Mult}}(\mathbf{h}, \mathbf{r}, \mathbf{t}) = (\mathbf{h} \circ \mathbf{t}) \cdot \mathbf{r}$$

$$f_{\mathrm{HolE}}(\mathbf{h}, \mathbf{r}, \mathbf{t}) = (\mathbf{h} \star \mathbf{t}) \cdot \mathbf{r}$$

- Hierarchy-aware embedding
  - Similar to CT, transform lower-level concepts into higher-level concepts

$$g_{\mathrm{HA}}(\mathbf{c}_h) = \sigma(\mathbf{W}_{\mathrm{HA}} \cdot \mathbf{c}_l + \mathbf{b}_{\mathrm{HA}})$$

The Joint Model

- Combine cross-view model and intra-view model

$$J = J_{\text{Intra}} + \omega \cdot J_{\text{Cross}}$$

- Where $J_{\text{Intra}} = J_{\text{Intra}}^{\mathcal{G}_I} + \alpha_1 \cdot J_{\text{Intra}}^{\mathcal{G}_O \setminus \mathcal{T}} + \alpha_2 \cdot J_{\text{Intra}}^{\text{HA}}$

Experiments

- Datasets
  - Constructed two new datasets from YAGO and DBpedia

Dataset	Instance Graph $\mathcal{G}_I$			Ontology Graph $\mathcal{G}_O$			Type Links $\mathcal{S}$
	#Entities	#Relations	#Triples	#Concepts	#Meta-relations	#Triples	
YAGO26K-906	26,078	34	390,738	906	30	8,962	9,962
DB111K-174	111,762	305	863,643	174	20	763	99,748

- Tasks
  - Triple completion
  - Entity typing
  - Ontology population
- Baselines: treat all links equally

Triple Completion

- CT is better than CG
- Hierarchy needs to be handled

Datasets	YAGO26K-906						DB111K-174					
Graphs	$\mathcal{G}_I$ KG Completion			$\mathcal{G}_O$ KG Completion			$\mathcal{G}_I$ KG Completion			$\mathcal{G}_O$ KG Completion		
Metrics	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
TransE (base)	0.195	14.09	34.51	0.145	12.29	20.59	0.327	22.26	49.01	0.313	23.22	46.91
TransE (all)	0.187	13.73	35.05	0.189	14.72	24.36	0.318	22.70	48.12	0.539	47.90	61.84
TransC	0.252	15.71	37.79	–	–	–	0.359	24.83	49.31	–	–	–
JOIE-TransE-CG	0.264	16.38	35.45	0.189	11.16	29.44	0.394	27.75	51.20	0.598	53.84	71.79
JOIE-TransE-CT	0.292	**18.72**	44.14	0.240	14.49	33.47	0.443	32.10	67.89	**0.622**	**58.10**	72.97
JOIE-HATransE-CT	**0.306**	18.62	**51.72**	**0.263**	**16.72**	**38.46**	**0.473**	**33.79**	**71.37**	0.591	52.07	**79.65**
DistMult (base)	0.253	22.91	28.76	0.197	**17.72**	25.08	0.265	25.95	27.63	0.235	15.18	29.11
DistMult (all)	0.288	**24.06**	31.24	0.156	14.32	16.54	0.280	27.24	29.70	0.501	45.52	64.73
JOIE-Mult-CG	0.274	18.80	37.45	0.198	11.16	27.91	0.320	23.44	49.49	0.532	46.15	68.91
JOIE-Mult-CT	**0.309**	20.40	**46.15**	**0.207**	14.71	30.43	**0.404**	**26.55**	**60.86**	**0.563**	**50.50**	71.62
JOIE-HAMult-CT	0.296	19.39	45.48	0.202	13.72	**31.10**	0.369	24.82	55.86	0.521	38.46	**77.25**
HolE (base)	0.265	**25.90**	28.31	0.192	18.70	20.29	0.301	29.24	31.51	0.227	18.91	32.83
HolE (all)	0.252	24.22	26.56	0.138	11.29	14.43	0.295	28.70	30.32	0.432	38.80	56.05
JOIE-HolE-CG	0.253	18.75	34.11	0.167	13.04	22.33	0.361	24.13	46.15	0.469	41.89	62.16
JOIE-HolE-CT	0.313	20.40	47.80	0.229	**20.85**	28.42	0.425	29.09	66.88	**0.514**	**43.24**	69.23
JOIE-HAHolE-CT	**0.327**	22.42	**52.41**	**0.236**	16.72	**30.96**	**0.464**	**33.11**	**69.56**	0.503	40.80	**71.03**

Entity Typing

- Significantly enhances the entity typing result

Datasets	YAGO26K-906			DB111K-174		
Metrics	MRR	Acc.	Hit@3	MRR	Acc.	Hit@3
TransE	0.144	7.32	35.26	0.503	43.67	60.78
MTransE	0.689	60.87	77.64	0.672	59.87	81.32
JOIE-TransE-CG	0.829	72.63	93.35	0.828	70.58	95.11
JOIE-TransE-CT	0.843	75.31	93.18	0.846	74.41	94.53
JOIE-HATransE-CT	**0.897**	**85.60**	**95.91**	**0.857**	**75.55**	**95.91**
DistMult	0.411	36.07	55.32	0.551	49.83	68.01
JOIE-Mult-CG	0.762	62.62	87.82	0.764	60.83	91.80
JOIE-Mult-CT	0.805	70.83	89.25	**0.791**	65.30	**93.47**
JOIE-HAMult-CT	**0.865**	**81.63**	**91.83**	0.778	**69.38**	85.71
HolE	0.395	34.83	54.79	0.504	44.75	65.38
JOIE-HolE-CG	0.777	65.30	87.89	0.784	66.75	89.37
JOIE-HolE-CT	0.813	72.27	88.71	0.805	68.84	**91.22**
JOIE-HAHolE-CT	**0.888**	**83.67**	**93.87**	**0.808**	**72.51**	89.79

Especially helpful for long-tail entities

- We select entities in YAGO26K-906 which occurs less than 8 times and entities in DB111K-174 which occurs less than 3 times.

Datasets	YAGO26K-906			DB111K-174		
Metrics	MRR	Acc.	Hit@3	MRR	Acc.	Hit@3
DistMult	0.156	10.89	25.33	0.219	16.48	33.71
MTransE	0.526	46.45	67.25	0.505	46.67	64.36
JOIE-TransE-CG	0.708	59.97	79.80	0.741	64.45	83.05
JOIE-TransE-CT	0.737	62.05	82.60	0.758	66.35	83.80
JOIE-HATransE-CT	**0.802**	**69.66**	**87.75**	**0.760**	**67.34**	**89.79**

Ontology Population

- Knowledge completion at ontology view

Query	Top 5 Populated Triples with distances
(scientist,?r, university)	scientist, ***graduated from***, university (0.499) scientist, ***isLeaderOf***, university (1.082) scientist, *isKnownFor*, university (1.098) scientist, *created*, university (1.119) scientist, ***livesIn***, university (1.141)
(boxer, ?r, club)	boxer, ***playsFor***, club (1.467) boxer, ***isAffiliatedTo***, club (1.474) boxer, ***worksAt***, club (1.479) boxer, *graduatedFrom*, club (1.497) boxer, *isConnectedTo*, club (1.552)

$$f_{\mathrm{CT}}^{\mathrm{inv}}(\mathbf{c}_{\mathrm{country}}) - f_{\mathrm{CT}}^{\mathrm{inv}}(\mathbf{c}_{\mathrm{office}}).$$

Outline

- Introduction
- Bringing First-Order Logic into Uncertain KG Embedding
- Bringing Ontological Concepts and Meta Relations into KG Embedding
- Summary & Future Work

Summary

- Logic Rules and PSL can help us to better handle uncertainty and incompleteness of KG
  - Chen et al., Embedding Uncertain Knowledge Graphs, AAAI'19
- Ontological View provides additional information for KG, where different types of links should be handled differently
  - Hao et al., Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts, KDD'19

Future Work

- How to automatically detect logic rules in KG?
- How to better leverage schema to conduct multi-hop reasoning?

PART III

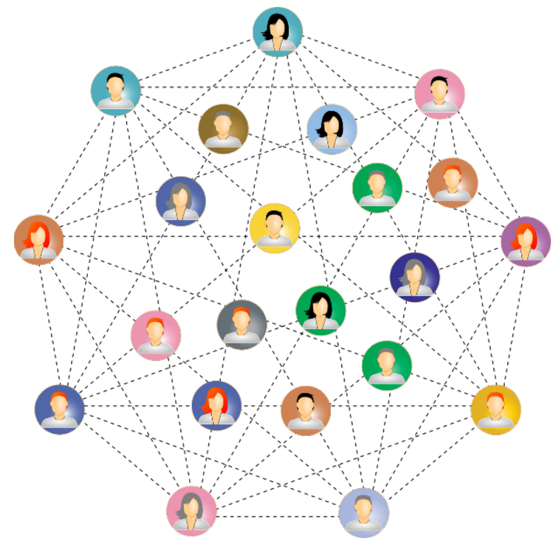
- Graph neural networks

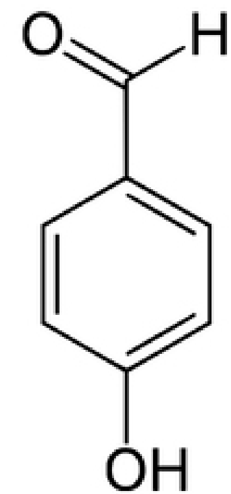
Outline

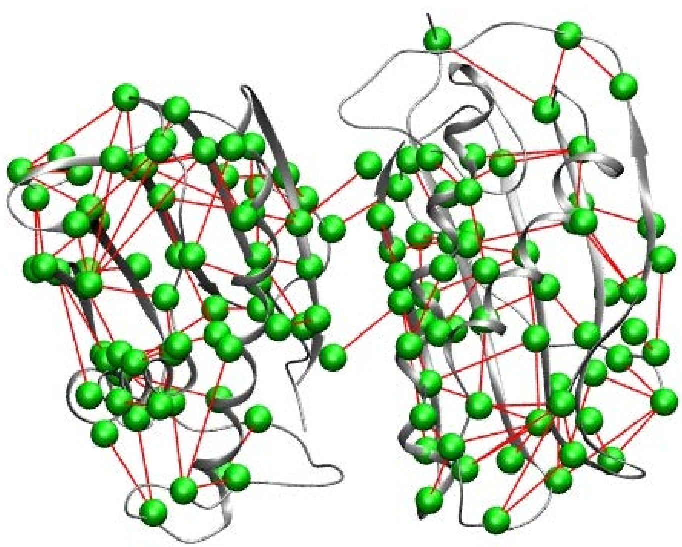
- Introduction
- Graph Neural Networks
- Graph Neural Networks for Heterogeneous Graphs
- Discussions

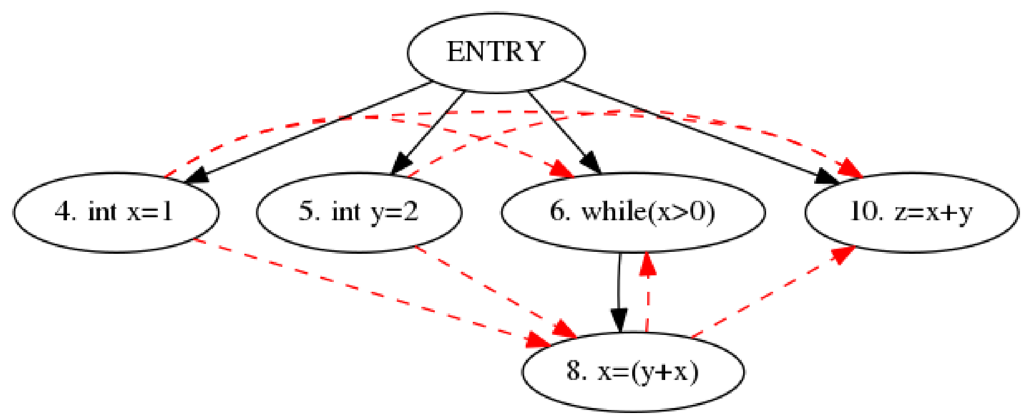
Graph Analysis

- Graphs are ubiquitous
  - Social networks
  - Proteins
  - Chemical compounds
  - Program dependence graph
  - …



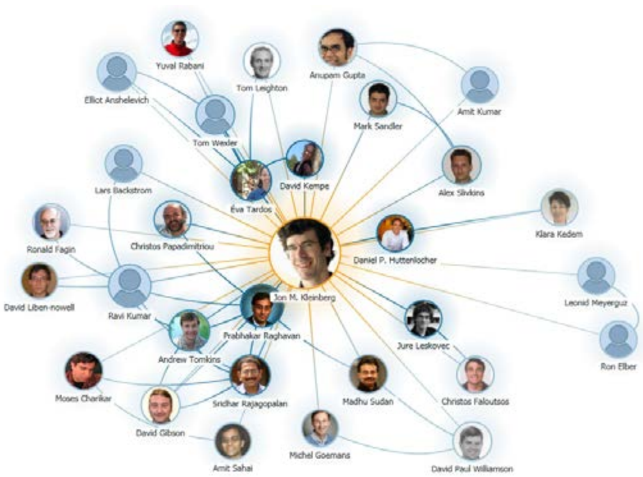






Representing Nodes and Graphs

- Important for many graph related tasks
- Discrete nature makes it very challenging
- Naïve solutions



	A	B	C	D	E	F
A	0	1	1	1	0	0
B	1	0	0	0	1	1
C	1	0	0	0	0	1
D	1	0	0	0	0	0
E	0	1	0	0	0	0
F	0	1	1	0	0	0

Limitations:

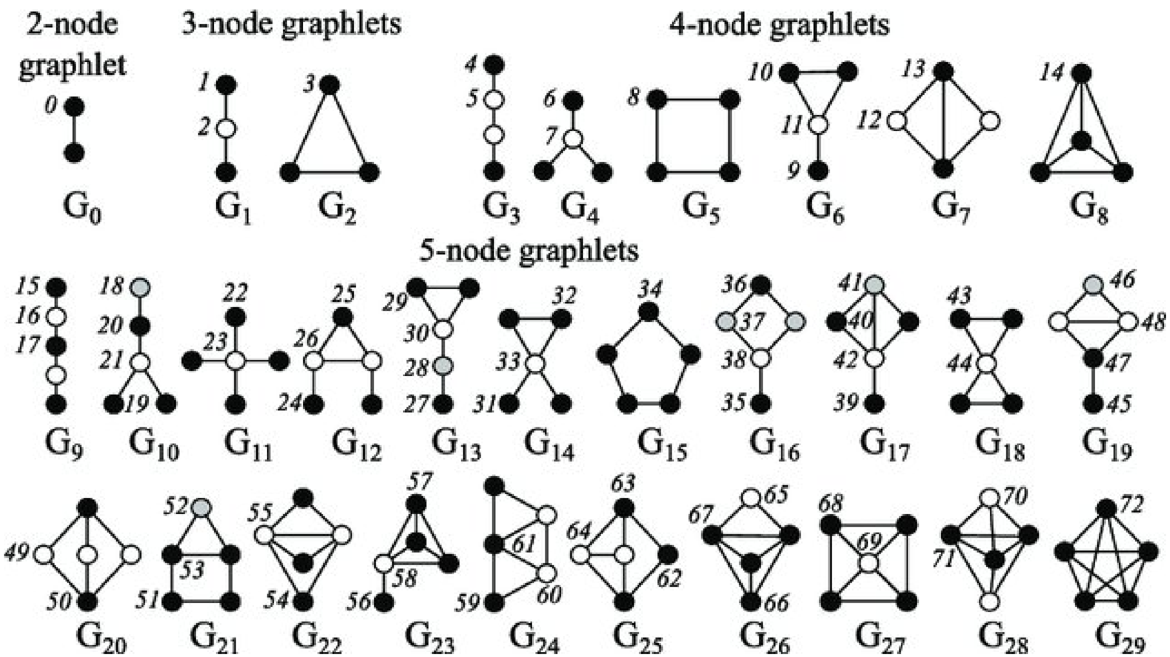
Extremely High-dimensional

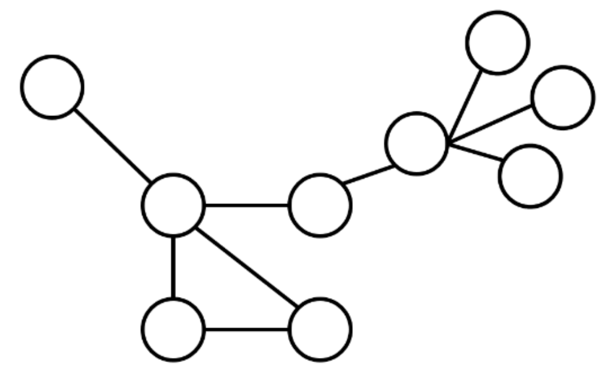
No global structure information integrated

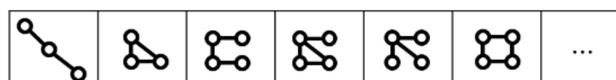
Permutation-variant

Even more challenging for graph representation

- Ex. Graphlet-based feature vector







12	1	4	1	6	0	...

Source: DOI: 10.1093/bioinformatics/btv130

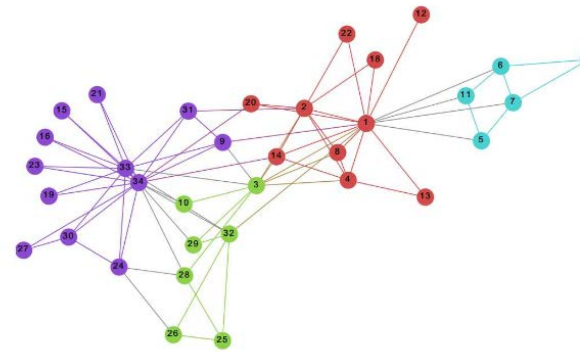
Requires subgraph isomorphism test: NP-hard

Source: https://haotang1995.github.io/projects/robust_graph_level_representation_learning_using_graph_based_structural_attentional_learning

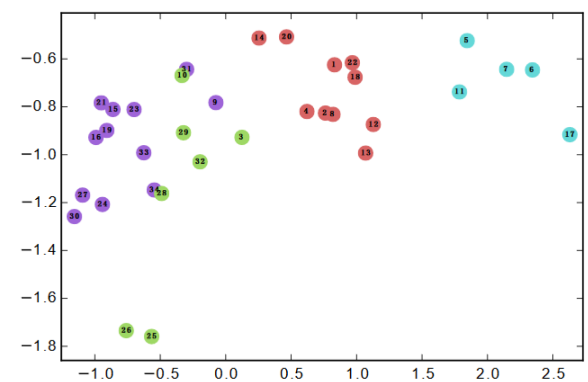
Automatic representation Learning

- Map each node/graph into a low dimensional vector
  - $\phi: V \rightarrow R^d$ or $\phi: \mathcal{G} \rightarrow R^d$
- Earlier methods
  - Shallow node embedding methods inspired by word2vec
    - DeepWalk [Perozzi, KDD'14]
    - LINE [Tang, WWW'15]
    - Node2Vec [Grover, KDD'16]

$\phi(v) = U^T x_v$, where U is the embedding matrix and x_v is the one-hot encoding vector



(a) Input: karate network



(b) Output: representations

Source: DeepWalk

Limitation of shallow embedding techniques

- Too many parameters
  - Each node is associated with an embedding vector, which are parameters
- Not inductive
  - Cannot handle new nodes
- Cannot handle node attributes

From shallow embedding to Graph Neural Networks

- The embedding function (encoder) is more complicated
  - Shallow embedding
    - $\phi(v) = U^T x_v$, where U is the embedding matrix and x_v is the one-hot encoding vector
  - Graph neural networks
    - $\phi(v)$ is a neural network depending on the graph structure

Outline

- Introduction
- Graph Neural Networks
- Graph Neural Networks for Heterogeneous Graphs
- Discussions

Notations

- An attributed graph $G = (V, E)$
  - V: vertex set
  - E: edge set
  - A: adjacency matrix
  - $X \in R^{d_0 \times |V|}$: feature matrix for all the nodes
  - $N(v)$: neighbors of node v
  - h_v^l: Representation vector of node v at Layer l
    - $Note\, h_v^0 = x_v$
  - $H^l \in R^{d_l \times |V|}$: representation matrix

The General Architecture of GNNs

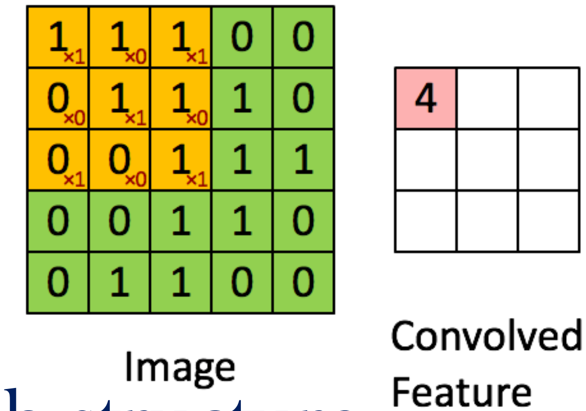
- For a node v at layer t

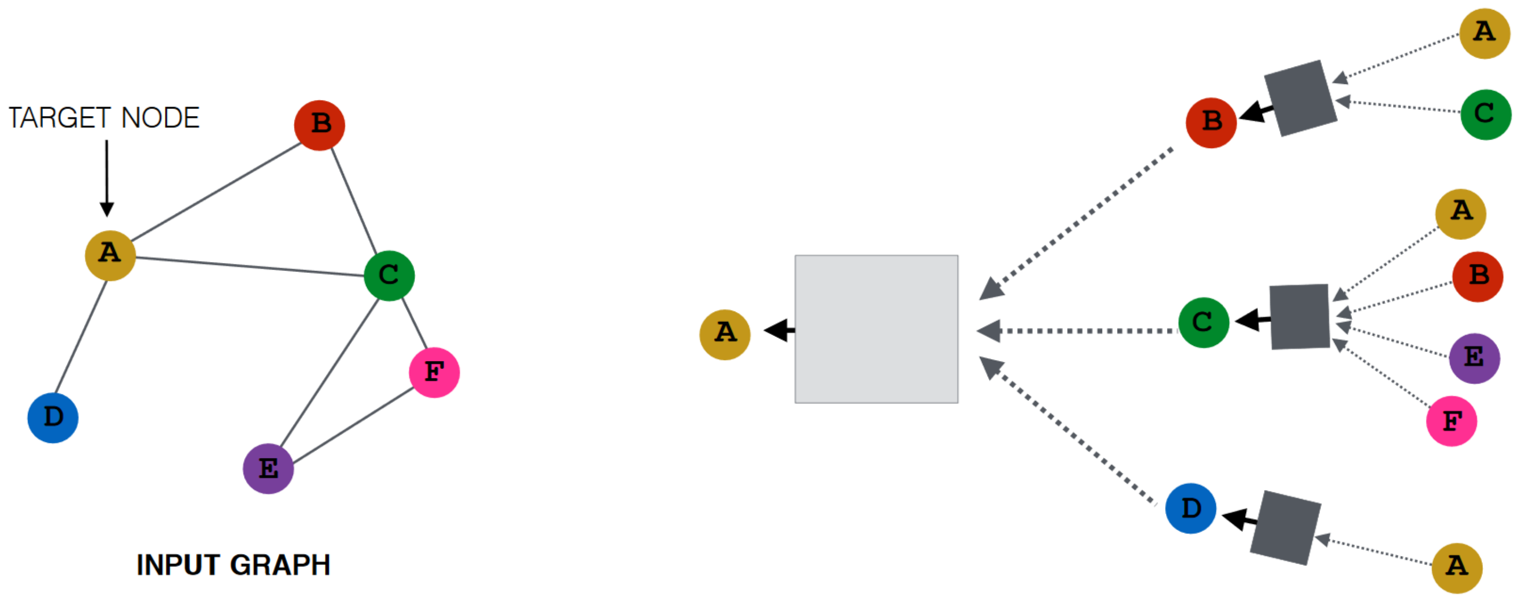
$$h_v^{(t)} = f\left(\underbrace{h_v^{(t-1)}}_{\text{representation vector from previous layer for node v}}, \left\{\underbrace{h_u^{(t-1)} | u \in \mathcal{N}(v)}_{\text{representation vectors from previous layer for node v's neighbors}}\right\}\right)$$

- A function of representations of neighbors and itself from previous layers
  - **Aggregation** of neighbors
  - **Transformation** to a different space
  - **Combination** of neighbors and the node itself

Compare with CNN

- Recall CNN
  - Regular graph
- GNN
  - Extend to irregular graph structure





Graph Convolutional Network (GCN)

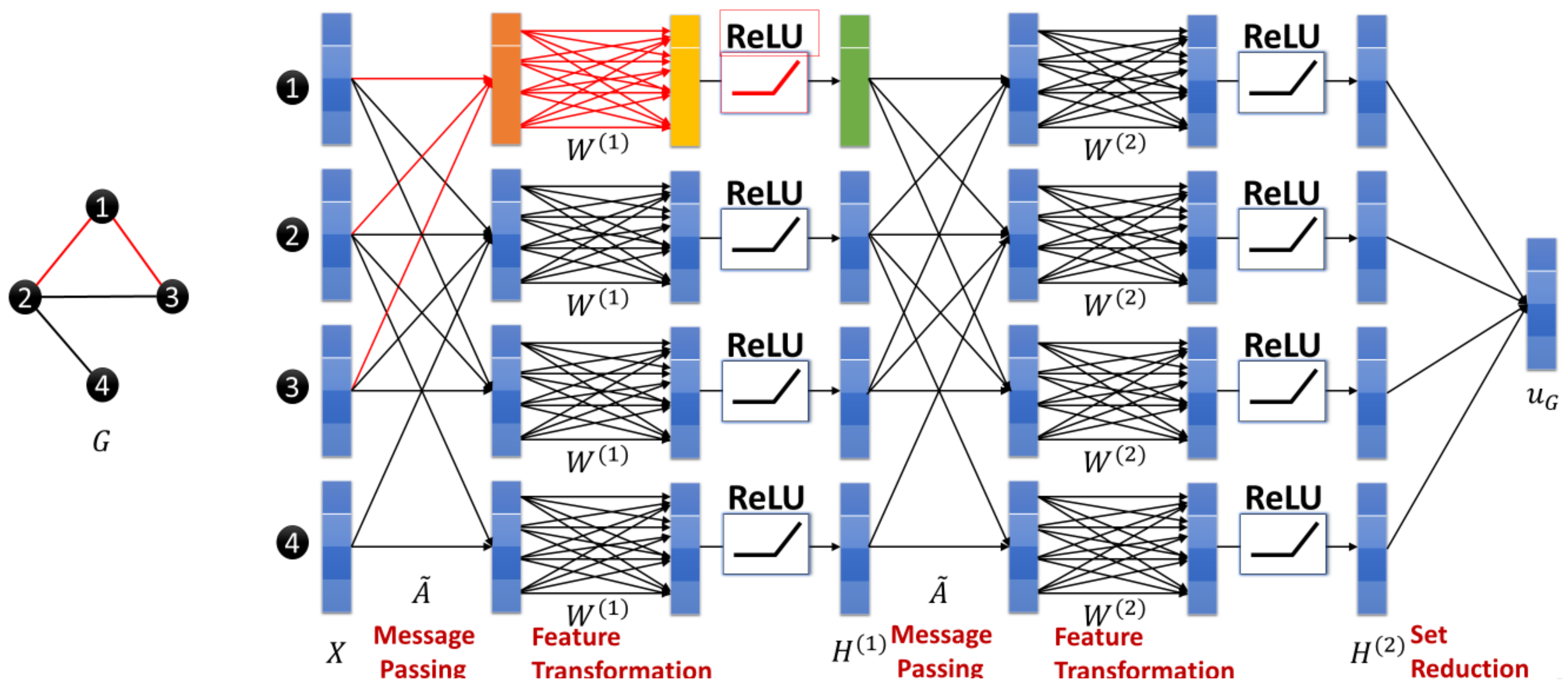
- Kipf and Welling, ICLR'17
  - $f(H^{(l)}, A) = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$, $\hat{A} = A + I$
  - f: graph filter
- From a node v's perspective

$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}_k \sum_{u \in N(v) \cup v} \frac{\mathbf{h}_u^{k-1}}{\sqrt{|N(u)||N(v)|}}\right)$$

W_k: *weight matrix at Layer k, shared across different nodes*

A toy example of 2-layer GCN on a 4-node graph

- Computation graph



GraphSAGE

- [Inductive Representation Learning on Large Graphs](#)
  William L. Hamilton*, Rex Ying*, Jure Leskovec, NeurIPS'17

$$\mathbf{h}^k_{\mathcal{N}(v)} \leftarrow \text{AGGREGATE}_k(\{\mathbf{h}^{k-1}_u, \forall u \in \mathcal{N}(v)\})$$

$$\mathbf{h}^k_v \leftarrow \sigma\left(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}^{k-1}_v, \mathbf{h}^k_{\mathcal{N}(v)})\right)$$

A more general form

$$\mathbf{h}^k_v = \sigma\left(\left[\mathbf{W}_k \cdot \boxed{\text{AGG}\left(\{\mathbf{h}^{k-1}_u, \forall u \in N(v)\}\right)}, \mathbf{B}_k \mathbf{h}^{k-1}_v\right]\right)$$

More about AGG

- Mean $$\mathrm{AGG} = \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|}$$
- LSTM $$\mathrm{AGG} = \mathrm{LSTM}\left([\mathbf{h}_u^{k-1}, \forall u \in \pi(N(v))]\right)$$
  - $\pi(\cdot)$: *a random permutation*
- Pool $$\mathrm{AGG} = \gamma \left\{\mathbf{Q}\mathbf{h}_u^{k-1}, \forall u \in N(v)\right\}$$
  - $\gamma(\cdot)$: Element-wise mean/max pooling of neighbor set

Message-Passing Neural Network

- Gilmer et al., 2017. Neural Message Passing for Quantum Chemistry. *ICML.*
- *A general framework that subsumes most GNNs*
  - Can also include edge information
- Two steps
  - Get messages from neighbors at step k

$$\mathbf{m}_v^k = \sum_{u \in N(v)} M(\mathbf{h}_u^{k-1}, \mathbf{h}_v^{k-1}, \mathbf{e}_{u,v})$$

e.g., Sum or MLP

  - Update the node latent represent based on the msg

$$\mathbf{h}_v^k = U(\mathbf{h}_v^{k-1}, \mathbf{m}_v^k)$$

e.g., LSTM, GRU

A special case: ***GGNN***, **Li et al., Gated graph sequence neural networks, ICLR 2015**

Graph Attention Network (GAN)

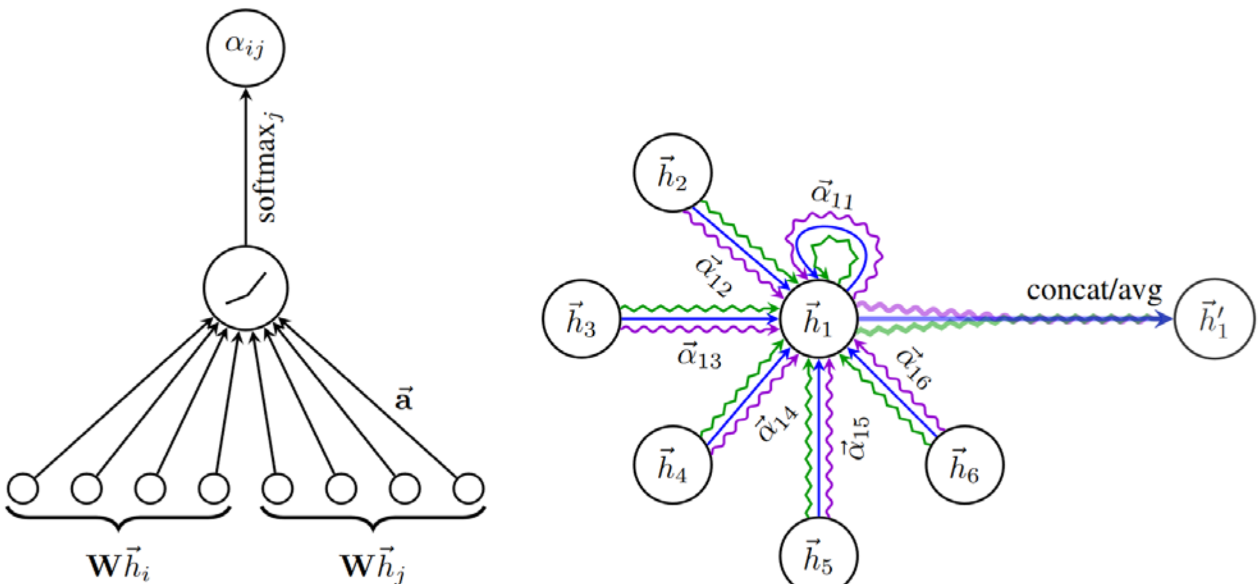
- How to decide the importance of neighbors?
  - GCN: a predefined weight
  - Others: no differentiation
- GAN: decide the weights using learnable attention
  - Velickovic et al., 2018. Graph Attention Networks. *ICLR.*

$$\vec{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \boxed{\alpha_{ij}} \mathbf{W} \vec{h}_j\right)$$

The attention mechanism

- Potentially many possible designs

$$\alpha_{ij} = \frac{\exp\left(\mathrm{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\mathrm{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)}$$

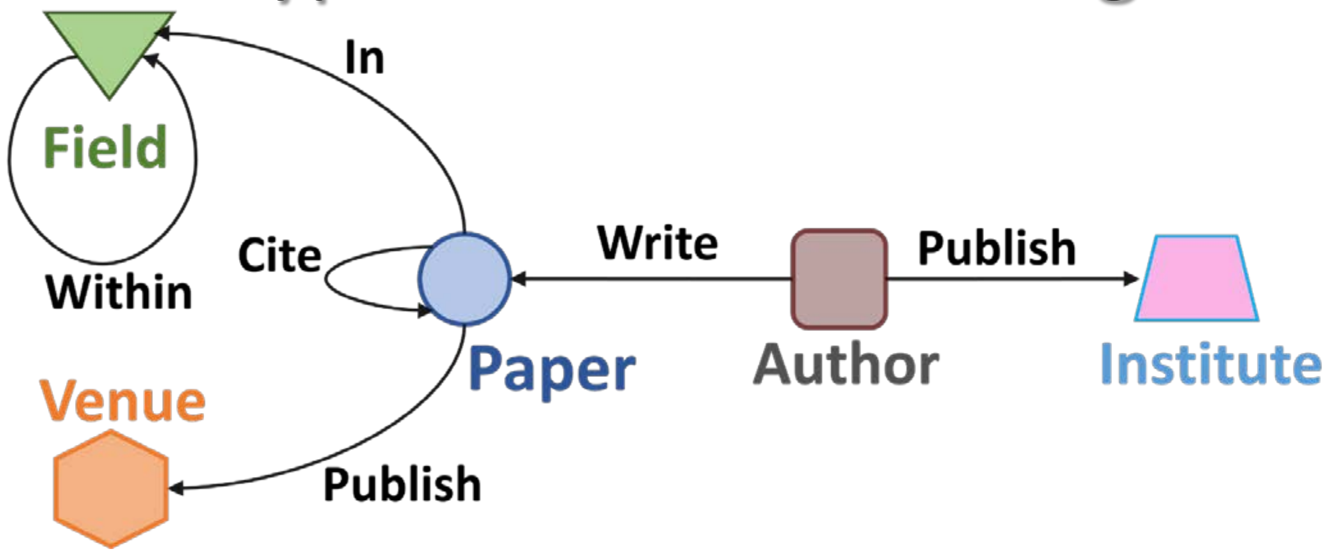


Outline

- Introduction
- Graph Neural Networks
- Graph Neural Networks for Heterogeneous Graphs
- Discussions

What are Heterogeneous Networks?

- Different types of nodes and edges



Example: Network Schema of Academic Networks

- Other examples:
  - E-Commerce
  - Knowledge graphs

Recap: GNNs

- Message passing framework

$$H^{l}[t] \leftarrow \underset{\forall s \in N(t), \forall e \in E(s,t)}{\textbf{Aggregate}} \left(\textbf{Extract}\left(H^{l-1}[s]; H^{l-1}[t], e \right) \right)$$

$Aggregate(\cdot)$: aggregate messages from different neighbors and edges

$Extract(\cdot)$: extract a message from $< t, e, s >$

- Attention scheme

$$H^{l}[t] \leftarrow \underset{\forall s \in N(t), \forall e \in E(s,t)}{\textbf{Aggregate}} \left(\textbf{Attention}(s,t) \cdot \textbf{Message}(s) \right)$$

$Attention(s,t)$: attention score on the edge $< s, t >$

$Message(s)$: information from s

Challenges Raised by HIN

- Message passing framework

$$H^l[t] \leftarrow \underset{\forall s \in N(t), \forall e \in E(s,t)}{\textbf{Aggregate}} \left(\textbf{Extract}\left(H^{l-1}[s]; H^{l-1}[t], e \right) \right)$$

- RGCN [ESWC'2018]: Parameterized by edge types
- HetGNN [WWW'19]: Parameterized by node types
- HAN [KDD'19]: Parameterized by meta-paths

Messages are of different types!

- Attention scheme

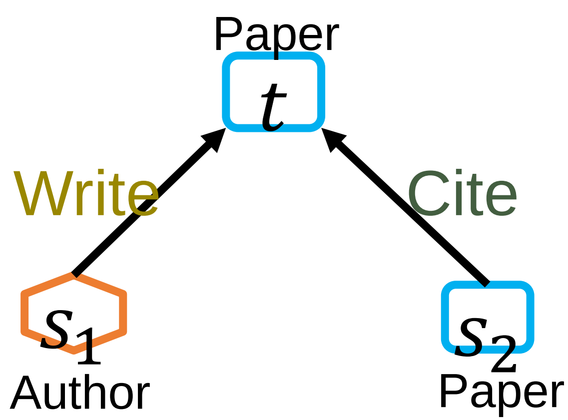
$$H^l[t] \leftarrow \underset{\forall s \in N(t), \forall e \in E(s,t)}{\textbf{Aggregate}} \left(\textbf{Attention}(s,t) \cdot \textbf{Message}(s) \right)$$

- HAN [KDD'19]: Attention weights parameterized by meta-paths

Our Solution: Heterogeneous Graph Transformer (HGT), Hu et al., WWW’20

- Parameterization by Meta-Relation
  - Meta-Relation: <source_type, edge_type, targe_type>
    - E.g., <author, first_author_of, paper>, <author, second_author_of, paper>
  - Parameter Sharing
    - Capture the correlation between different meta-relations
    - More efficient in terms of parameter space

$W_{\langle \text{Author}, \text{Write}, \text{Paper} \rangle}$
$= W_{\text{Author}} \; W_{\text{Write}} \; W_{\text{Paper}}$



$W_{\langle \text{Paper}, \text{Cite}, \text{Paper} \rangle}$
$= W_{\text{Paper}} \; W_{\text{Cite}} \; W_{\text{Paper}}$

Message, Attention, and Aggregation of HGT

- Heterogeneous message for an edge <s, e, t>

$$\mathbf{Message}_{HGT}(s,e,t) = \underset{i\in[1,h]}{\Big\|} MSG\text{-}head^i(s,e,t)$$

$$MSG\text{-}head^i(s,e,t) = \text{M-Linear}^i_{\tau(s)}\left(H^{(l-1)}[s]\right) W^{MSG}_{\phi(e)}$$

- Heterogeneous mutual attention on edge <s, e, t>

$$\mathbf{Attention}_{HGT}(s,e,t) = \underset{\forall s\in N(t)}{\text{Softmax}}\left(\underset{i\in[1,h]}{\Big\|} ATT\text{-}head^i(s,e,t)\right) \quad (3)$$

$$ATT\text{-}head^i(s,e,t) = \left(K^i(s)\, W^{ATT}_{\phi(e)}\, Q^i(t)^T\right) \cdot \boxed{\frac{\mu_{\langle\tau(s),\phi(e),\tau(t)\rangle}}{\sqrt{d}}}$$

$$K^i(s) = \text{K-Linear}^i_{\tau(s)}\left(H^{(l-1)}[s]\right)$$

$$Q^i(t) = \text{Q-Linear}^i_{\tau(t)}\left(H^{(l-1)}[t]\right)$$

Significant score for each meta-relation

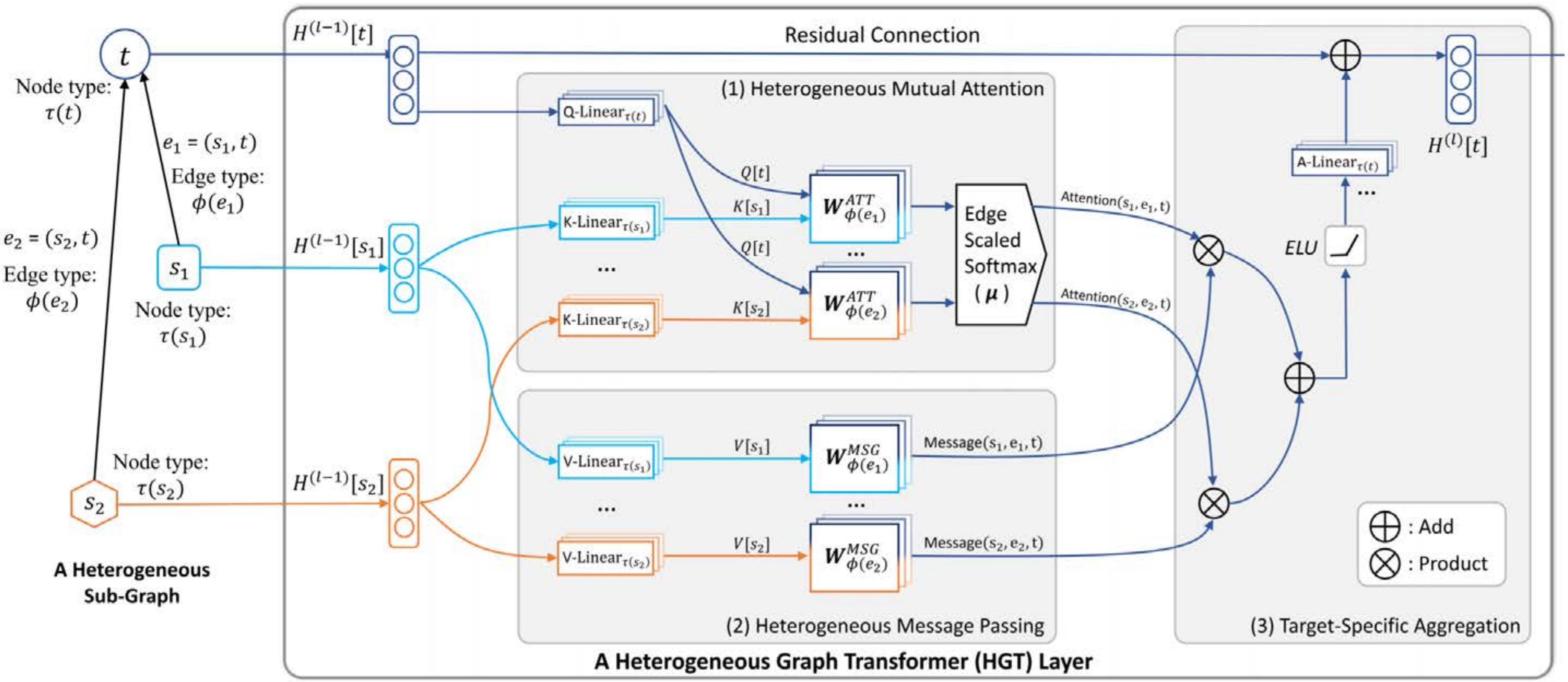
- Task-specific aggregation

$$\widetilde{H}^{(l)}[t] = \underset{\forall s\in N(t)}{\bigoplus}\left(\mathbf{Attention}_{HGT}(s,e,t)\cdot \mathbf{Message}_{HGT}(s,e,t)\right)$$

$$H^{(l)}[t] = \text{A-Linear}_{\tau(t)}\left(\sigma\left(\widetilde{H}^{(l)}[t]\right)\right) + H^{(l-1)}[t].$$

Architecture of HGT

- Putting together



Leaderboard #1 on Open Graph Benchmark

Leaderboard for ogbn-mag

The classification accuracy on the test set. The higher, the better.

Package: >=1.2.1

Rank	Method	Accuracy	Contact	References	#Params	Hardware	Date
1	**HGT (LADIES Sample)**	0.5007 ± 0.0043	Ziniu Hu	Paper, Code	21,173,389	Tesla K80 (12GB GPU)	Jul 7, 2020
2	GraphSAINT (R-GCN aggr)	0.4751 ± 0.0022	Matthias Fey – OGB team	Paper, Code	154,366,772	GeForce RTX 2080 (11GB GPU)	Jun 26, 2020
3	NeighborSampling (R-GCN aggr)	0.4678 ± 0.0067	Matthias Fey – OGB team	Paper, Code	154,366,772	GeForce RTX 2080 (11GB GPU)	Jun 26, 2020
4	Full-batch R-GCN	0.3977 ± 0.0046	Matthias Fey – OGB team	Paper, Code	154,366,772	Quadro RTX 8000 (48GB GPU)	Jun 26, 2020
5	ClusterGCN (R-GCN aggr)	0.3732 ± 0.0037	Matthias Fey – OGB team	Paper, Code	154,366,772	GeForce RTX 2080 (11GB GPU)	Jun 26, 2020

Leaderboard for ogbn-products

The classification accuracy on the test set. The higher, the better.

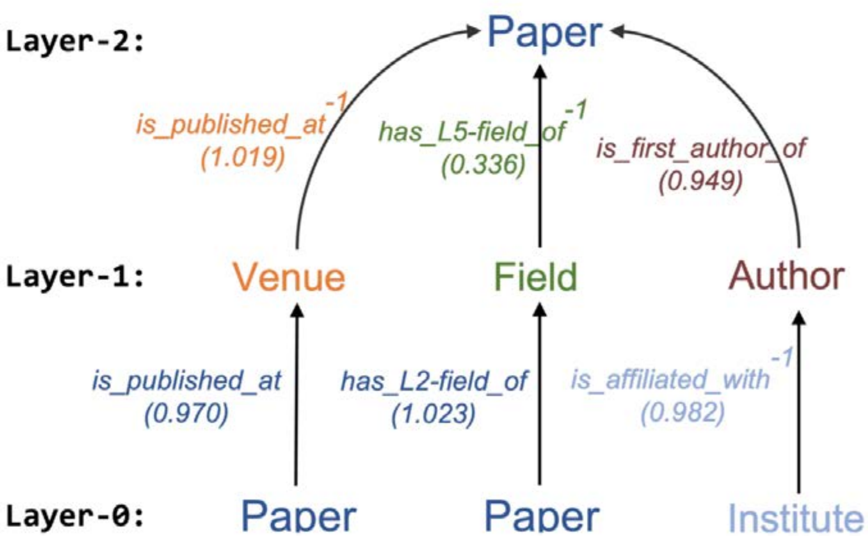
Package: >=1.1.1

Rank	Method	Accuracy	Contact	References	#Params	Hardware	Date
1	**HGT (LADIES Sample)**	0.8560 ± 0.0040	Ziniu Hu	Paper, Code	2,025,573	Tesla K80 (12GB GPU)	Jul 8, 2020
2	**DeeperGCN**	0.8098 ± 0.0020	Guohao Li - DeepGCNs.org	Paper, Code	253,743	NVIDIA Tesla V100 (32GB GPU)	Jun 28, 2020
3	GAT with NeighborSampling	0.7945 ± 0.0059	Matthias Fey	Paper, Code	1,751,574	GeForce RTX 2080 (11GB GPU)	May 24, 2020
4	GraphSAINT (SAGE aggr)	0.7908 ± 0.0024	Matthias Fey – OGB team	Paper, Code	206,895	GeForce RTX 2080 (11GB GPU)	Jun 10, 2020
5	ClusterGCN (SAGE aggr)	0.7897 ± 0.0033	Matthias Fey – OGB team	Paper, Code	206,895	GeForce RTX 2080 (11GB GPU)	Jun 10, 2020
6	NeighborSampling (SAGE aggr)	0.7870 ± 0.0036	Matthias Fey – OGB team	Paper, Code	206,895	GeForce RTX 2080 (11GB GPU)	Jun 10, 2020
7	Full-batch GraphSAGE	0.7850 ± 0.0014	Matthias Fey – OGB team	Paper, Code	206,895	Quadro RTX 8000 (48GB GPU)	Jun 20, 2020
8	GraphSAGE	0.7829 ± 0.0016	Quan Gan (DGL Team)	Paper, Code	Please tell us	Please tell us	May 12, 2020

Case Studies

Venue	Time	Top−5 Most Similar Venues
WWW	2000	SIGMOD, VLDB, NSDI, GLOBECOM, SIGIR
	2010	GLOBECOM, KDD, CIKM, SIGIR, SIGMOD
	2020	KDD, GLOBECOM, SIGIR, WSDM, SIGMOD
KDD	2000	SIGMOD, ICDE, ICDM, CIKM, VLDB
	2010	ICDE, WWW, NeurIPS, SIGMOD, ICML
	2020	NeurIPS, SIGMOD, WWW, AAAI, EMNLP
NeurIPS	2000	ICCV, ICML, ECCV, AAAI, CVPR
	2010	ICML, CVPR, ACL, KDD, AAAI
	2020	ICML, CVPR, ICLR, ICCV, ACL

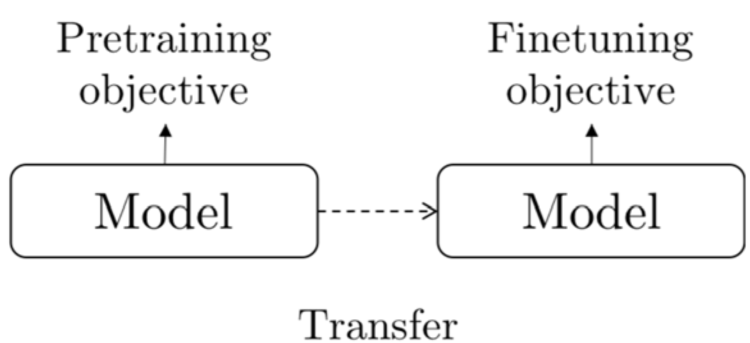
- Conferences' topics changed over time.
- The relative temporal encoding can help capture this temporal evolution.



- HGT can implicitly extract meta paths for specific downstream tasks, without manual customization.
  - Read from $\mu\langle\tau(s),\phi(e),\tau(t)\rangle$

Pre-Training of Graph Neural Networks

- Challenges on training GNNs
  - Requires abundant task-specific labeled data
- What is pre-training?
  - Train GNNs with self-supervision and then transfer learned model to downstream tasks with only a few labels
  - Popular in NLP: e.g., BERT

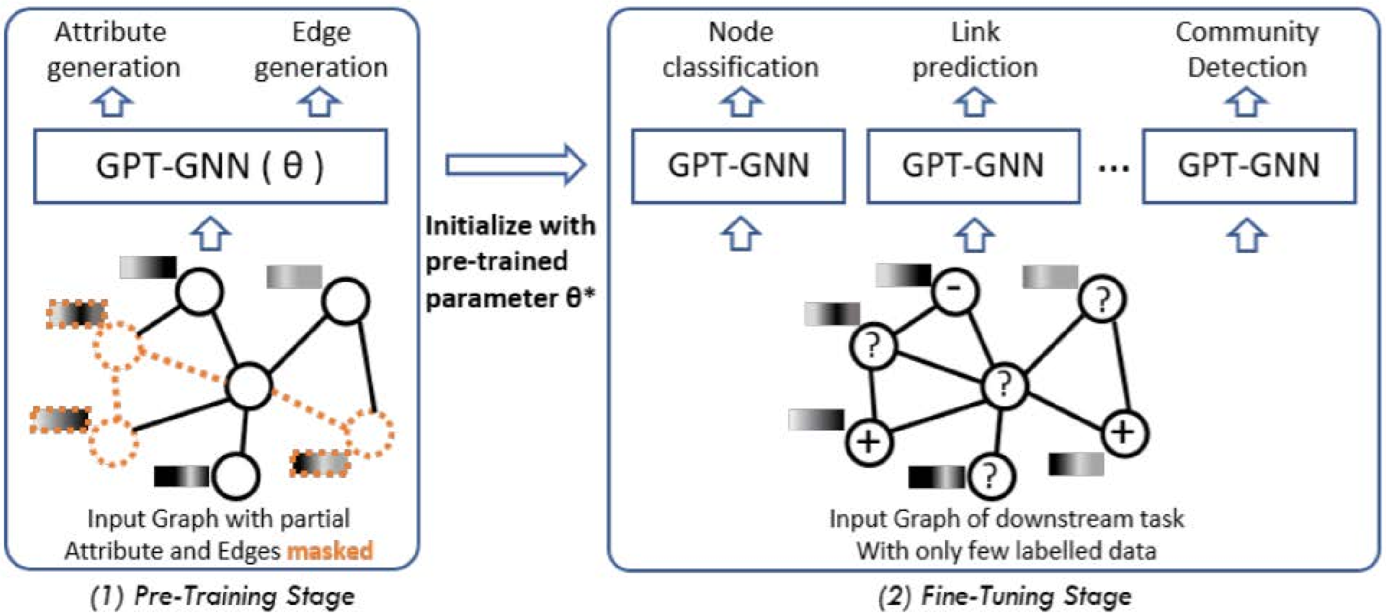


Key to the Success of Pre-Training

- Self-supervised Tasks
  - No additional labels are needed
  - General enough to different downstream tasks
- Existing self-supervised tasks for graphs
  - Link prediction [GAE, GraphSAGE (NIPS'17)]
  - Maximize mutual information between a patch and its super graph [DGI (ICLR'19)]

Our Solution: GPT-GNN (Hu et al., KDD'20)

- Pre-train GNNs via the generative task to generate both node attributes and edges
  - Goal: find GNN parameters $\theta^* = \max_\theta p(G; \theta)$



Model $p(G; \theta)$

- Average over different node order permutation π

$$p(G;\theta) = \mathbb{E}_{\pi}\left[p_\theta(X^\pi, E^\pi)\right]$$

X: node attributes; E: edge list for all nodes

- Factorize the joint probability autoregressively given π

$$\log p_\theta(X^\pi, E^\pi) = \sum_{i=1} \log p_\theta(X_i^\pi, E_i^\pi \mid X_{<i}^\pi, E_{<i}^\pi).$$

X_i: attribute for node i; E_i: edge list for node i

- Factorize the conditional probability $p(current|past)$

$E_{i,\neg o}$: unobserved edges for node i;
$E_{i,o}$: observed edges for node i

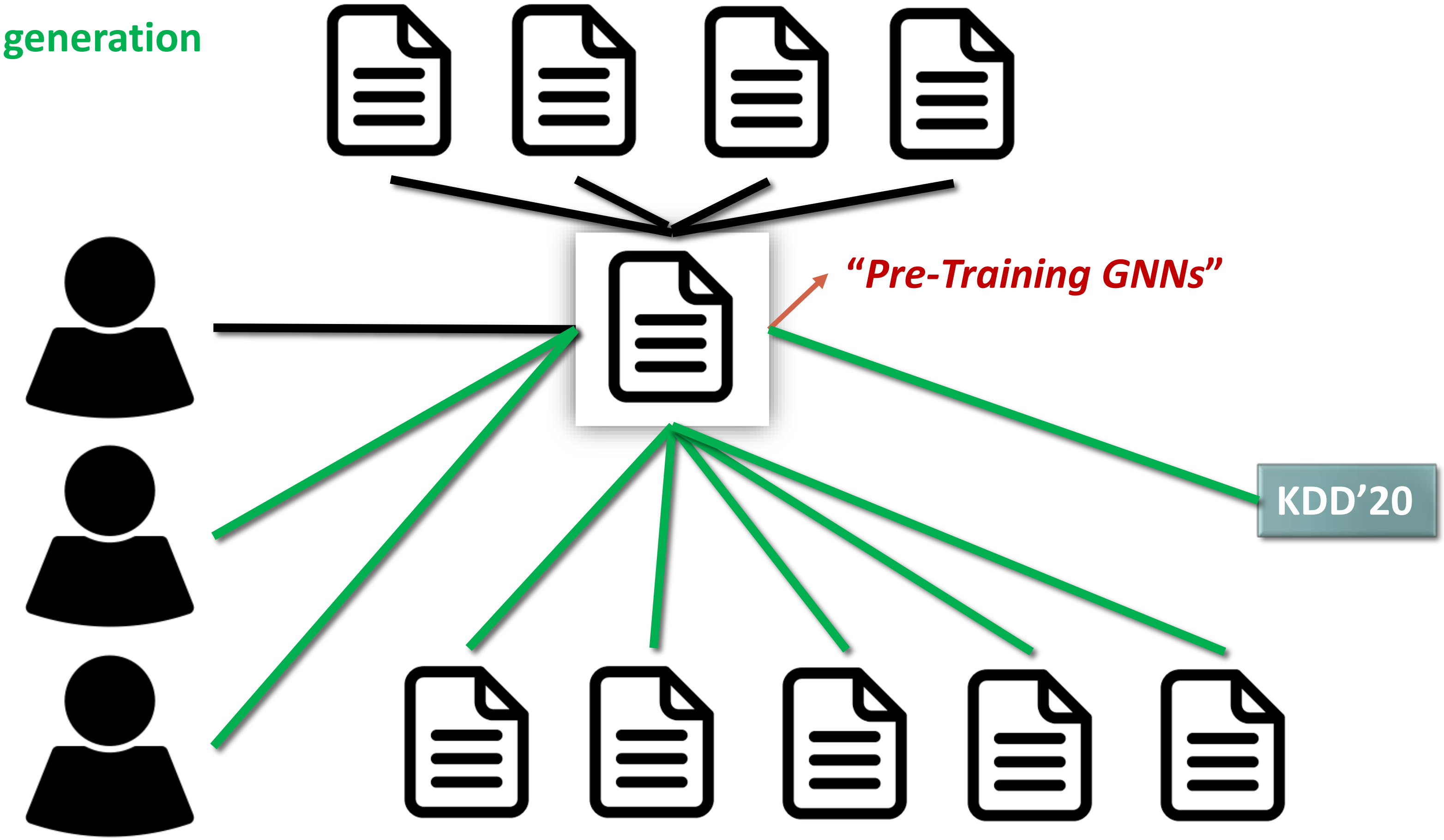
$$\begin{aligned}
&p_\theta(X_i^\pi, E_i^\pi \mid X_{<i}^\pi, E_{<i}^\pi) \\
&= \sum_o p_\theta(X_i^\pi, E_{i,\neg o}^\pi \mid E_{i,o}^\pi, X_{<i}^\pi, E_{<i}^\pi) \cdot p_\theta(E_{i,o}^\pi \mid X_{<i}^\pi, E_{<i}^\pi) \\
&= \mathbb{E}_o\left[p_\theta(X_i^\pi, E_{i,\neg o}^\pi \mid E_{i,o}^\pi, X_{<i}^\pi, E_{<i}^\pi)\right] \\
&= \mathbb{E}_o\Big[\underbrace{p_\theta(X_i^\pi \mid E_{i,o}^\pi, X_{<i}^\pi, E_{<i}^\pi)}_{\text{1) generate attributes}} \cdot \underbrace{p_\theta(E_{i,\neg o}^\pi \mid E_{i,o}^\pi, X_{\leq i}^\pi, E_{<i}^\pi)}_{\text{2) generate edges}}\Big].
\end{aligned}$$

Illustration of the Factorization

Observed links

Attribute generation

Link generation



Results

- Data 1: Open Academic Graph

	Pre-Train	Fine-Tune
No Transfer:	CS Academic Graph	CS Academic Graph
Field Transfer:	Med, Bio, Physics…	CS Academic Graph
Time Transfer:	CS before 2014	CS after 2014
Time + Field Transfer:	Med, Bio, Physics… before 2014	CS after 2014

- **All pre-training frameworks** help the performance of GNNs
  - GAE, GraphSage, Graph Infomax
  - GPT-GNN
- **GPT-GNN helps the most** by achieving a relative performance gain of 9.1% over the base model without pre-training
- **Both self-supervised tasks in GPT-GNN** help the pre-training framework
  - Attribute generation
  - Edge generation

	Downstream Dataset	OAG		
	Evaluation Task	Paper–Field	Paper–Venue	Author ND
	No Pre-train	.336±.149	.365±.122	.794±.105
Field Transfer	GAE	.403±.114	.418±.093	.816±.084
	GraphSAGE (unsp.)	.368±.125	.401±.096	.803±.092
	Graph Infomax	.387±.112	.404±.097	.810±.084
	GPT-GNN (Attr)	.396±.118	.423±.105	.818±.086
	GPT-GNN (Edge)	.401±.109	.428±.096	.826±.093
	GPT-GNN	**.407±.107**	**.432±.098**	**.831±.102**
Time Transfer	GAE	.384±.117	.412±.101	.812±.095
	GraphSAGE (unsp.)	.352±.121	.394±.105	.799±.093
	Graph Infomax	.369±.116	.398±.102	.805±.089
	GPT-GNN (Attr)	.382±.114	.414±.098	.811±.089
	GPT-GNN (Edge)	.392±.105	.421±.102	.821±.088
	GPT-GNN	**.400±.108**	**.429±.101**	**.825±.093**
Time + Field Transfer	GAE	.371±.124	.403±.108	.806±.102
	GraphSAGE (unsp.)	.349±.130	.393±.118	.797±.097
	Graph Infomax	.360±.121	.391±.102	.800±.093
	GPT-GNN (Attr)	.364±.115	.409±.103	.809±.094
	— (w/o node separation)	.347±.128	.391±.102	.791±.108
	GPT-GNN (Edge)	.386±.116	.414±.104	.815±.105
	— (w/o adaptive queue)	.376±.121	.410±.115	.808±.104
	GPT-GNN	**.393±.112**	**.420±.108**	**.818±.102**

Outline

- Introduction
- Graph Neural Networks
- Graph Neural Networks for Heterogeneous Graphs
- Discussions

Open Questions

- Why GNNs work?
  - Is the nonlinear transformation necessary?
  - Chen et al., Are Powerful Graph Neural Nets Necessary? A Dissection on Graph Classification, arXiv:1905.04579
  - A concatenate feature vector from graph propagation, followed by a MLP works equally well, and much faster!

$$X^G = \gamma(G, X) = \left[\boldsymbol{d}, X, \tilde{A}^1 X, \tilde{A}^2 X, \cdots, \tilde{A}^K X\right],$$

Q & A

- Thanks to my collaborators:
  - Junheng Hao, Xuelu (Shirley) Chen, Ziniu Hu, Kewei (Vivian) Cheng, Wei Wang, Kai-Wei Chang, Carlo Zanialo, Muhao Chen, Yuxiao Dong, Kuansan Wang, etc…

- Thanks to my funding agencies and industry support:
  - NSF, DARPA, PPDAI, Yahoo!, Nvidia, Snapchat, Amazon, Okawa Foundation