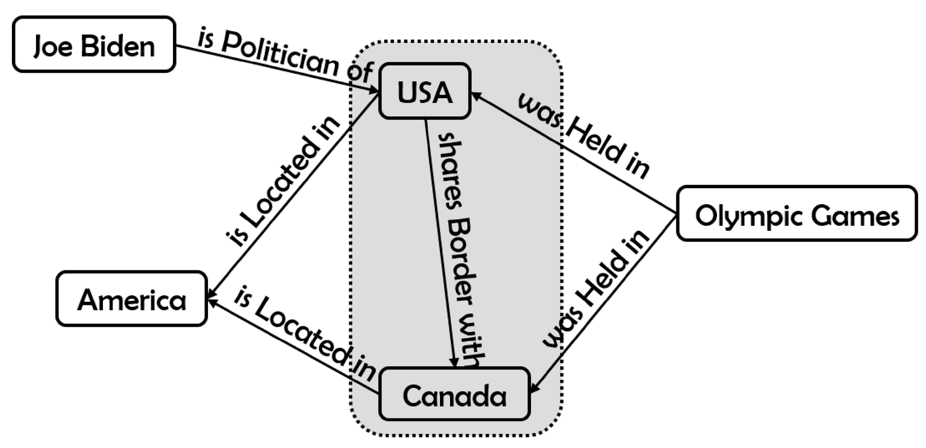
Knowledge Graph Embedding via Metagraph Learning

Chanyoung Chung and Joyce Jiyoung Whang
School of Computing, KAIST

The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021

Preliminaries & Intuition

- A knowledge graph is a graphical representation of human knowledge.
  - Each fact is represented as a triplet (head entity, relation, tail entity).
  - Knowledge graph embedding is a representation learning technique that projects entities and relations into a continuous feature space.
- Intuition: Semantic closeness can be inferred by the structural similarity between entities. If two entities share the same head or tail entity with the same relation, they might belong to the same semantic category.



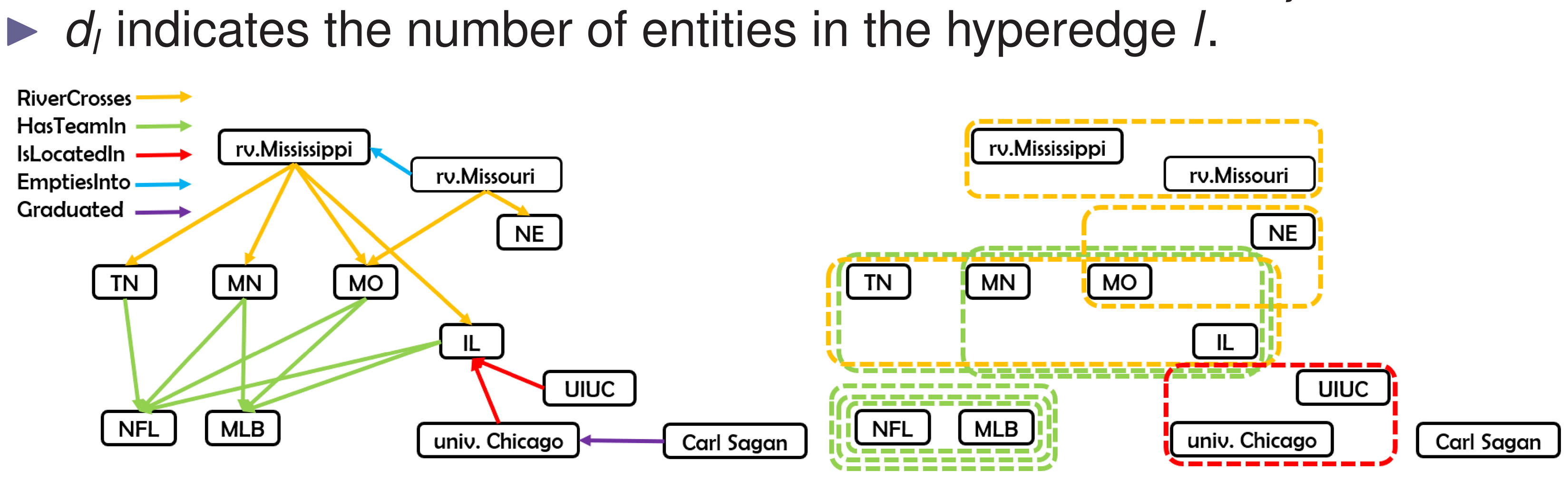
- 'USA' and 'Canada' are structurally similar.
  - They share the same tail entity 'America' with the relation 'is Located in'.
  - They share the same head entity 'Olympic Games' with the relation 'was Held in'.

Main Contributions

- Propose a new affinity metric that measures the structural similarity between entities by converting a knowledge graph into a hypergraph.
- Define the metagraph of a knowledge graph by grouping semantically close entities and extracting representative interactions between entities.
- Propose the metagraph-based pre-training model of knowledge graph embedding which is effective in improving the accuracy of state-of-the-art knowledge graph embedding methods.

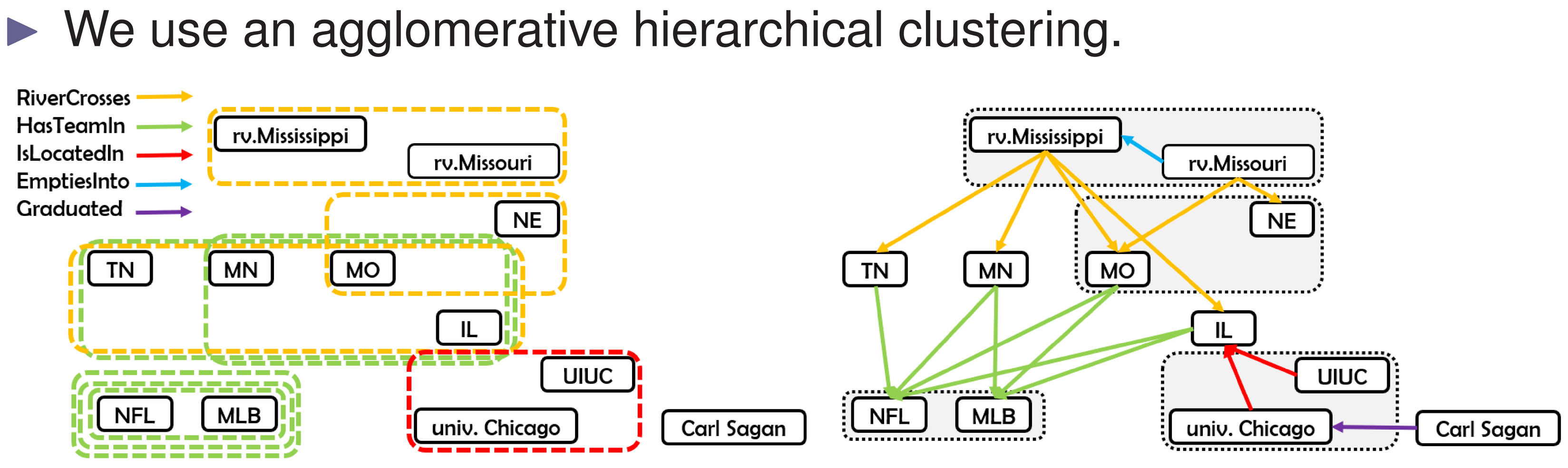
Step 1: Hypergraph Representation of a Knowledge Graph

- Define a new affinity score that reflects the structural similarity.
  - Connect a set of entities via a hyperedge if they share the same head entity (or the same tail entity) with the same relation.
- Affinity score between entities v_i and v_j is defined as $a_{ij} = \sum_{l \in \mathcal{L}} 1/d_l^2$.
  - $\mathcal{L}$ indicates the set of hyperedges which contain v_i and v_j simultaneously.
  - d_l indicates the number of entities in the hyperedge l.



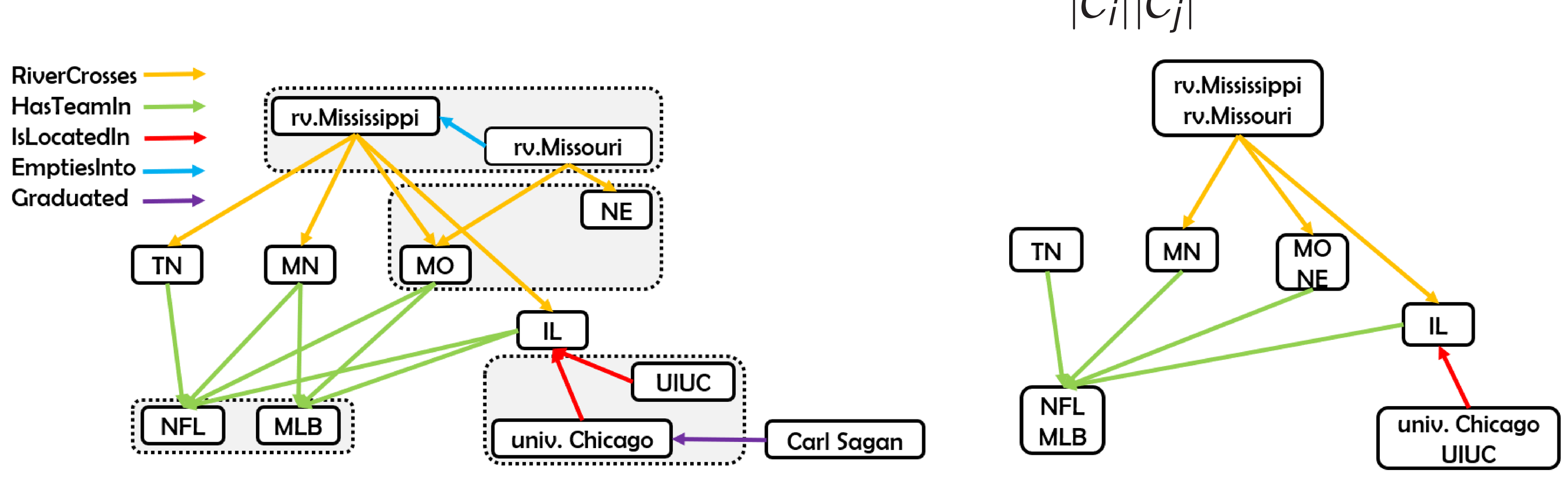
Step 2: Grouping Entities

- Normalize the affinity scores: $\hat{a}_{ij} = \frac{a_{ij}}{\sum_k a_{ik}} + \frac{a_{ij}}{\sum_k a_{kj}}$
- Group similar entities by hypergraph clustering with the normalized scores.
  - We use an agglomerative hierarchical clustering.



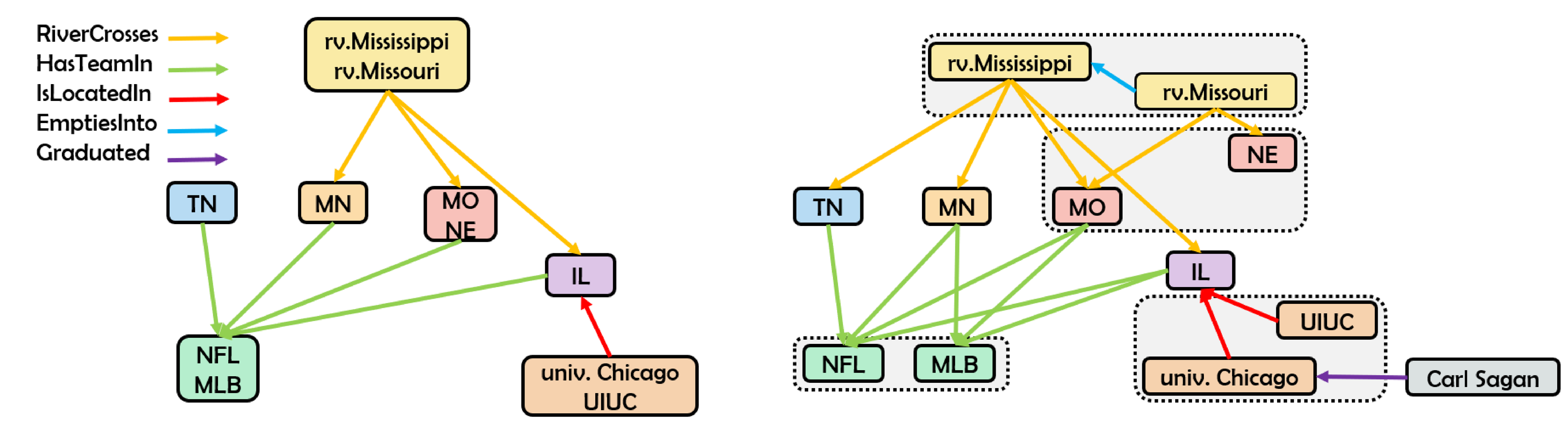
Step 3: Metagraph of a Knowledge Graph

- Construct the metagraph of a knowledge graph.
  - The metagraph preserves a core structure of a given knowledge graph.
  - Merge entities in the same group to form a super-entity.
  - Within-group triplets are dropped.
  - Given two super entities $\mathcal{C}_i$ and $\mathcal{C}_j$, we add a triplet $(\mathcal{C}_i, r, \mathcal{C}_j)$ to the metagraph with the probability of $\frac{|\{(h,r,t) | h \in \mathcal{C}_i \wedge t \in \mathcal{C}_j \wedge r \in \mathcal{R}\}|}{|\mathcal{C}_i||\mathcal{C}_j|}$.



Step 4. Pre-training of Knowledge Graph Embedding

- Run a knowledge graph embedding method on the metagraph.
- Initialize the corresponding entities and relations in the original knowledge graph with the learned representations on the metagraph.
  - Entities in the same group are initialized with the same representations.



Experimental Results on Affinity Scores

- Top 5 most similar entities to the target entity in NELL-995.
  - Our affinity measure successfully detects semantically close entities.

Target entity	Top 5 most similar entities to the target entity (ties are all included)
emotion_thankfulness	emotion_graditude, emotion_admiration, emotion_happiness, emotion_joy, emotion_deep_love, emotion_jealousy, emotion_thanks
software_microsoft_word	software_internet_explorer, software_microsoft_frontpage, software_microsoft_powerpoint, software_notepad, software_autocad
sport_american_football	sport_ski, sport_scout, sport_skiing, sport_golf, sport_judo
university_harvard	university_harvard_university, university_harvard_law, school_oxford, university_harvard_law_school, university_john_f__kennedy_school
furniture_queen_bed	furniture_queen, furniture_king_beds, furniture_king_size_beds, furniture_twin_beds, furniture_queen_size_beds

Experimental Results on Link Prediction

- Link prediction results on three benchmark datasets
  - Gain is calculated by

$$\text{Gain}_{\text{metric}} = sign(\text{metric}) \frac{(Score_{\text{model}} - Score_{\text{meta-model}})}{Score_{\text{model}}} \times 100\%$$

where $sign(\text{MR}) = 1$ and $sign(\text{MRR}) = sign(\text{Hit@10}) = -1$.
  - Our metagraph-based pre-training always shows positive total gains.
  - Our method is effective in improving the performance of the knowledge graph embedding methods.

		MR (↓)	MRR (↑)	Hit@10 (↑)	Total Gain (↑)
FB15K	TransE	89.0	**0.596**	0.733	
	meta-TransE	**75.0**	0.551	**0.798**	
	Gain (↑)	**15.8%**	-7.5%	**8.8%**	**17.1%**
	DistMult	**106.4**	0.414	0.644	
	meta-DistMult	143.7	**0.541**	**0.786**	
	Gain (↑)	-35.0%	**30.8%**	**21.9%**	**17.7%**
	RotatE	34.4	**0.691**	0.869	
	meta-RotatE	**33.6**	0.690	**0.871**	
	Gain (↑)	**2.1%**	-0.1%	**0.2%**	**2.2%**
NELL-995	TransE	7202.4	0.278	**0.477**	
	meta-TransE	**6507.5**	**0.287**	0.434	
	Gain (↑)	**9.6%**	**3.3%**	-9.1%	**3.8%**
	DistMult	10312.7	**0.298**	0.388	
	meta-DistMult	**8046.0**	0.288	**0.397**	
	Gain (↑)	**22.0%**	-3.6%	**2.3%**	**20.7%**
	RotatE	9243.9	0.350	0.428	
	meta-RotatE	**8618.7**	**0.352**	**0.435**	
	Gain (↑)	**6.8%**	**0.7%**	**1.8%**	**9.3%**
WN18	TransE	210.4	0.521	0.943	
	meta-TransE	**185.9**	**0.535**	**0.949**	
	Gain (↑)	**11.6%**	**2.7%**	**0.7%**	**15.0%**
	DistMult	301.1	0.320	0.550	
	meta-DistMult	**289.1**	**0.463**	**0.732**	
	Gain (↑)	**4.0%**	**44.7%**	**33.2%**	**81.9%**
	RotatE	76.681	**0.661**	0.882	
	meta-RotatE	**73.718**	0.655	**0.884**	
	Gain (↑)	**3.9%**	-0.9%	**0.2%**	**3.2%**

Conclusion & Future Work

- We propose the metagraph-based pre-training method for knowledge graph embedding by proposing a new affinity metric that measures the structural similarity between entities in a knowledge graph.
- We plan to extend our method to generate overlapping clusters of entities. Also, we can easily extend our method to a multi-level framework.