InGram: Inductive Knowledge Graph Embedding via Relation Graphs

▼ BDILab ▼ GitHub

Jaejun Lee, **Chanyoung Chung**, and **Joyce Jiyoung Whang***

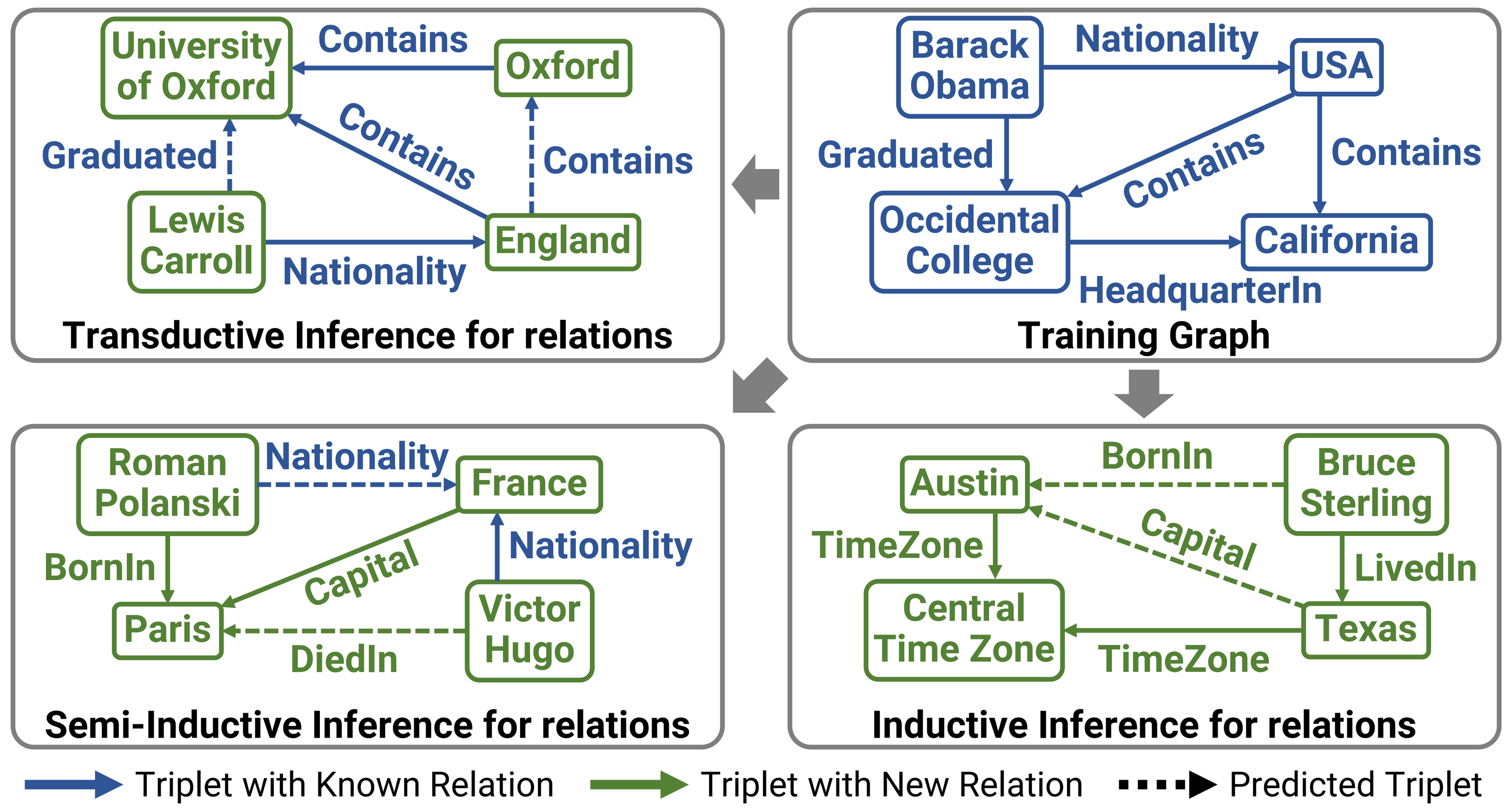
School of Computing, KAIST

The 40th International Conference on Machine Learning (ICML 2023)

Main Contributions

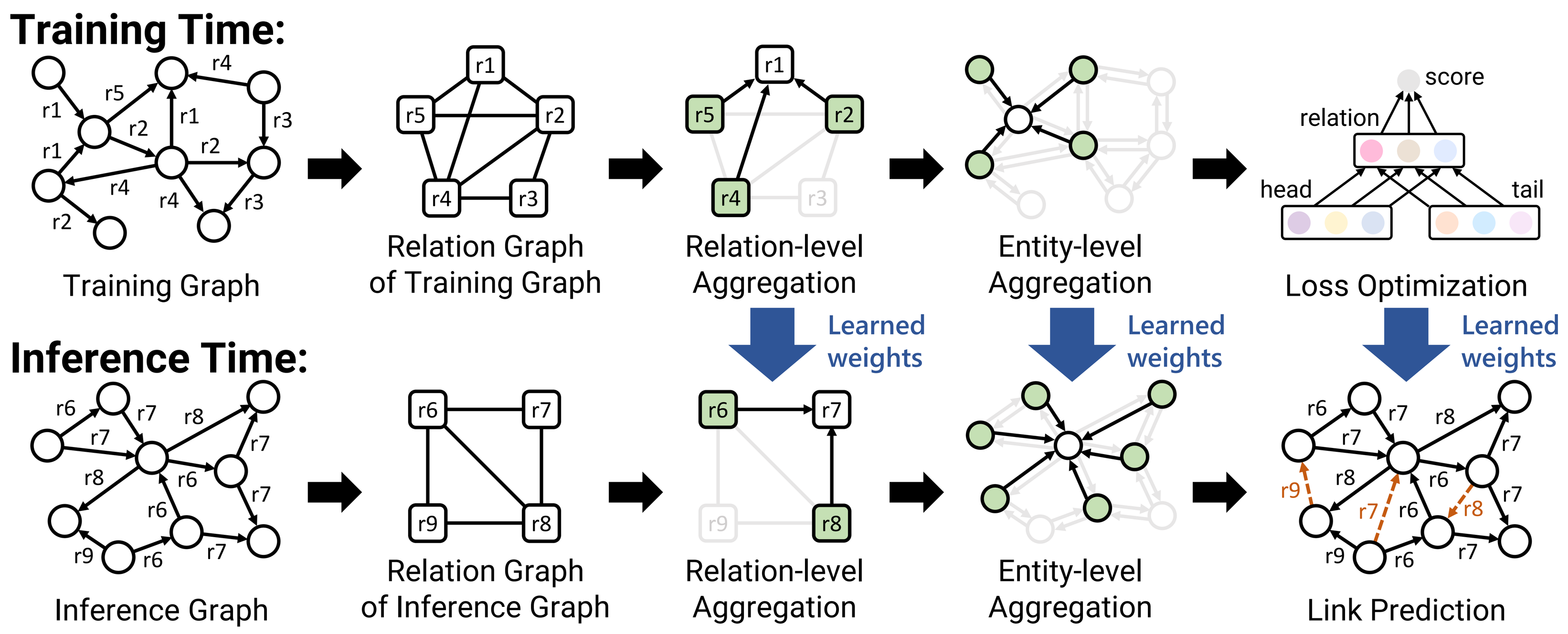
- Propose **InGram**, an **In**ductive knowledge **Gra**ph e**m**bedding method, that can generate embedding vectors for **new relations and entities**
  - Based on the **structural similarities between relations**, define the **relation graph** to designate neighboring relations for each relation
  - Learn **how to aggregate** neighboring embeddings to generate relation and entity embeddings using an attention mechanism
  - Introduce **dynamic split** and **re-initialization** that makes InGram more easily **generalizable** to a new graph
- Generate 13 real-world datasets; InGram significantly outperforms 14 different state-of-the-art methods in **inductive link prediction with varied ratios of new relations**

Inductive Learning Scenarios



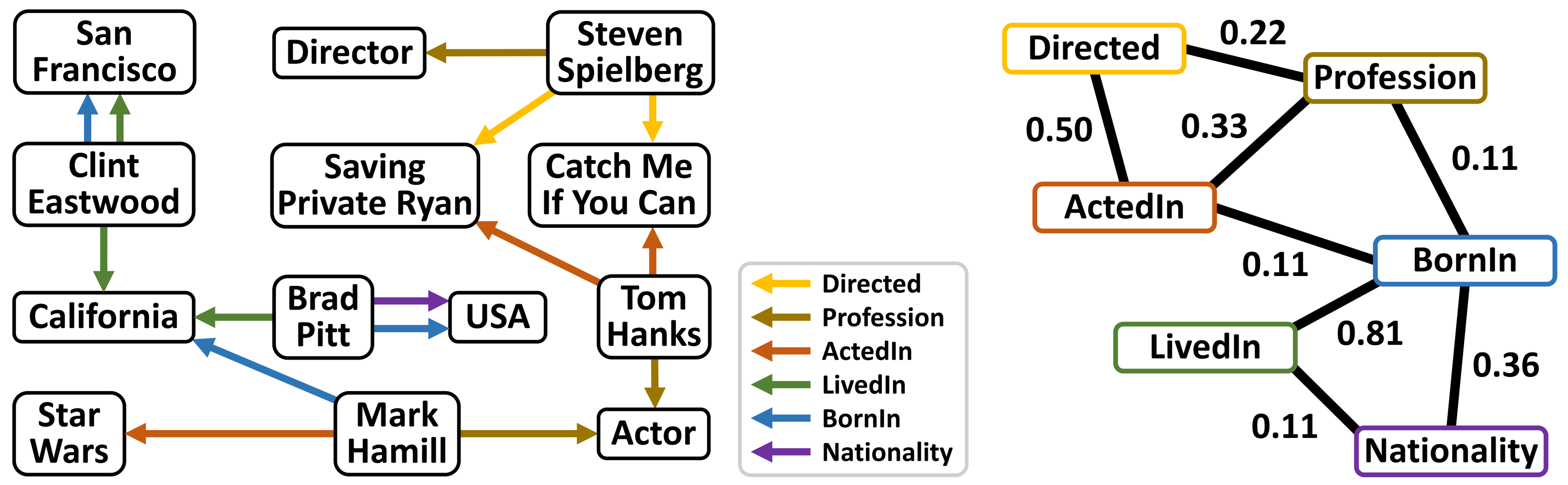
- **Semi-Inductive Inference for relations**
  - An inference graph contains both **known** and **new relations**
- **Inductive Inference for relations**
  - **All relations** are **new** in an inference graph

Overview of InGram



Relation Graph

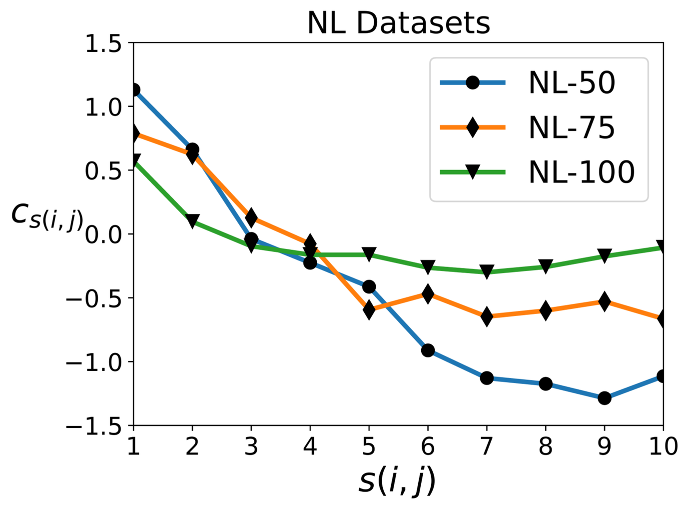
- Define the **neighboring relations** of each relation
  - Each node corresponds to **a relation**
  - Each edge weight indicates **the affinity between two relations**
- Adjacency matrix of the relation graph $\boldsymbol{A} = \boldsymbol{E}_h^{\top}\boldsymbol{D}_h^{-2}\boldsymbol{E}_h + \boldsymbol{E}_t^{\top}\boldsymbol{D}_t^{-2}\boldsymbol{E}_t$
  - Consider **how many entities are shared** between two relations and **how frequently they share the same entity**



Knowledge Graph — Relation Graph

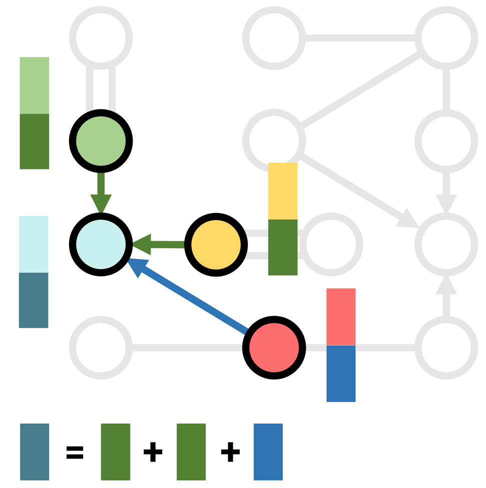
Relation-level Aggregation

- Aggregate neighboring relations' embedding vectors
  - $\mathbf{z}_i^{(l+1)} = \sigma\left(\sum_{r_j\in\mathcal{N}_i}\alpha_{ij}^{(l)}\boldsymbol{W}^{(l)}\mathbf{z}_j^{(l)}\right)$
- Consider the **relative importance** and the **affinity weight**
  - $\alpha_{ij}^{(l)} = \dfrac{\exp\left(\psi^{(l)}\left(\left[\mathbf{z}_i^{(l)}\|\mathbf{z}_j^{(l)}\right]\right) + c_{s(i,j)}^{(l)}\right)}{\sum_{r_{j'}\in\mathcal{N}_i}\exp\left(\psi^{(l)}\left(\left[\mathbf{z}_i^{(l)}\|\mathbf{z}_{j'}^{(l)}\right]\right) + c_{s(i,j')}^{(l)}\right)}$
  - $\psi^{(l)}(\mathbf{x}) = \mathbf{y}^{(l)}\sigma(\boldsymbol{P}^{(l)}\mathbf{x})$



Entity-level Aggregation

- Compute an entity embedding by considering its **own vector**, its **neighbors' embeddings**, and its **adjacent relations**
  - $\mathbf{h}_i^{(l+1)} = \sigma\left(\beta_{ii}^{(l)}\widehat{\boldsymbol{W}}^{(l)}\left[\mathbf{h}_i^{(l)}\|\bar{\mathbf{z}}_i^{(L)}\right] + \sum_{v_j\in\widehat{\mathcal{N}}_i}\sum_{r_k\in\mathcal{R}_{ji}}\beta_{ijk}^{(l)}\widehat{\boldsymbol{W}}^{(l)}\left[\mathbf{h}_j^{(l)}\|\mathbf{z}_k^{(L)}\right]\right)$
  - $\bar{\mathbf{z}}_i^{(L)} = \sum_{v_j\in\widehat{\mathcal{N}}_i}\sum_{r_k\in\mathcal{R}_{ji}}\dfrac{\mathbf{z}_k^{(L)}}{\sum_{v_{j'}\in\widehat{\mathcal{N}}_i}|\mathcal{R}_{j'i}|}$
  - $\beta_{ii}^{(l)} = \exp\left(\hat{\psi}^{(l)}\left(\left[\mathbf{h}_i^{(l)}\|\mathbf{h}_i^{(l)}\|\bar{\mathbf{z}}_i^{(L)}\right]\right)\right)/\lambda$
  - $\beta_{ijk}^{(l)} = \exp\left(\hat{\psi}^{(l)}\left(\left[\mathbf{h}_i^{(l)}\|\mathbf{h}_j^{(l)}\|\mathbf{z}_k^{(L)}\right]\right)\right)/\lambda$
  - $\hat{\psi}^{(l)}(\mathbf{x}) = \hat{\mathbf{y}}^{(l)}\sigma(\widehat{\boldsymbol{P}}^{(l)}\mathbf{x})$
  - $\lambda = \exp\left(\hat{\psi}^{(l)}\left(\left[\mathbf{h}_i^{(l)}\|\mathbf{h}_i^{(l)}\|\bar{\mathbf{z}}_i^{(L)}\right]\right)\right) + \sum_{v_{j'}\in\widehat{\mathcal{N}}_i}\sum_{r_{k'}\in\mathcal{R}_{j'i}}\exp\left(\hat{\psi}^{(l)}\left(\left[\mathbf{h}_i^{(l)}\|\mathbf{h}_{j'}^{(l)}\|\mathbf{z}_{k'}^{(L)}\right]\right)\right)$



Modeling Relation-Entity Interactions & Training Regime

- Final embedding vectors: $\mathbf{z}_k = \boldsymbol{M}\mathbf{z}_k^{(L)}$ and $\mathbf{h}_i = \widehat{\boldsymbol{M}}\mathbf{h}_i^{(\widehat{L})}$
- Scoring function: $f(v_i, r_k, v_j) = \mathbf{h}_i^{\top}\mathrm{diag}(\overline{\boldsymbol{W}}\mathbf{z}_k)\mathbf{h}_j$
- Use **margin-based ranking loss** to optimize the model parameters
- **Dynamic split**: Randomly **re-split** the fact set and the training set
- **Re-initialization**: Randomly **re-initialize** all feature vectors

Experimental Results

- Datasets: **13 real-world datasets** with various inductive settings
- **Baselines**: GraIL, CoMPILE, SNRI, INDIGO, RMPI, BLP, QBLP, RAILD, NeuralLP, DRUM, NBFNet, RED-GNN, CompGCN, NodePiece
- **Link Prediction Results: Inductive Inference for Relations**

		MR (↓)	MRR (↑)	Hit@10 (↑)	Hit@1 (↑)
NL-100	Best-baseline	143.9	0.220	0.385	0.136
	InGram	**92.6**	**0.309**	**0.506**	**0.212**
WK-100	Best-baseline	2005.6	0.096	0.136	0.070
	InGram	**1515.7**	**0.107**	**0.169**	**0.072**
FB-100	Best-baseline	375.6	0.121	0.263	0.053
	InGram	**171.5**	**0.223**	**0.371**	**0.146**

- **Link Prediction Results: Semi-Inductive Inference for Relations**

		MR (↓)	MRR (↑)	Hit@10 (↑)	Hit@1 (↑)
NL-75	Best-baseline	242.5	0.203	0.361	0.129
	InGram	**59.1**	**0.261**	**0.464**	**0.167**
WK-75	Best-baseline	523.9	0.172	0.290	0.110
	InGram	**315.5**	**0.247**	**0.362**	**0.179**
FB-75	Best-baseline	705.1	0.107	0.201	0.057
	InGram	**217.4**	**0.189**	**0.325**	**0.119**

- **Link Prediction Results: Transductive Inference for Relations**

		MR (↓)	MRR (↑)	Hit@10 (↑)	Hit@1 (↑)
NL-0	Best-baseline	160.2	0.263	0.430	0.177
	InGram	**152.4**	**0.269**	**0.431**	**0.189**
NELL-995-v1	Best-baseline	7.1	0.677	0.885	0.550
	InGram	**6.0**	**0.739**	**0.895**	**0.660**

Conclusion & Future Work

- Define the **relation graph** to handle new relations at inference time
- **InGram** learns to generate embeddings for **new relations and entities** solely based on the structure of a given knowledge graph
- InGram significantly outperforms 14 different baseline methods on **inductive**, **semi-inductive**, and **transductive inferences for relations**
- We will explore the **theoretical analysis** of InGram and make InGram **robust to possibly noisy information** in a given knowledge graph

* Corresponding Author (jjwhang@kaist.ac.kr) | GitHub: https://github.com/bdi-lab/InGram | Lab Homepage: https://bdi-lab.kaist.ac.kr