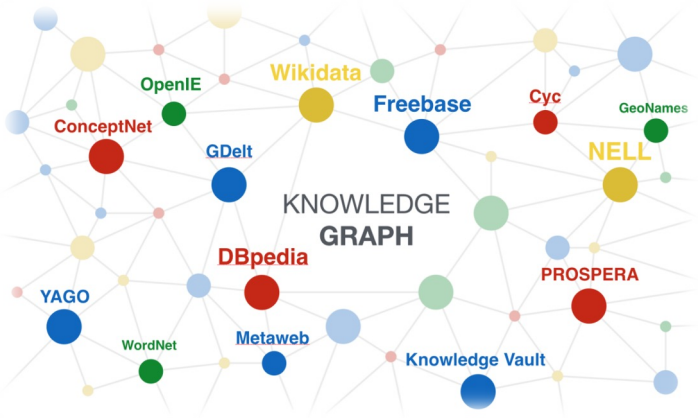
R-GCN:

Modeling Relational Data with Graph Convolutional Networks



CS886 Paper Presentation

Presented by **Moein Shirdel**

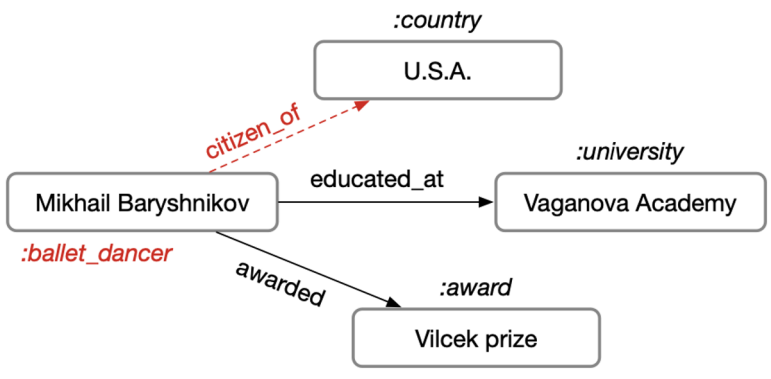
Outline

1. Introduction
2. Background
3. Methodology
4. Results
5. Conclusion

Introduction

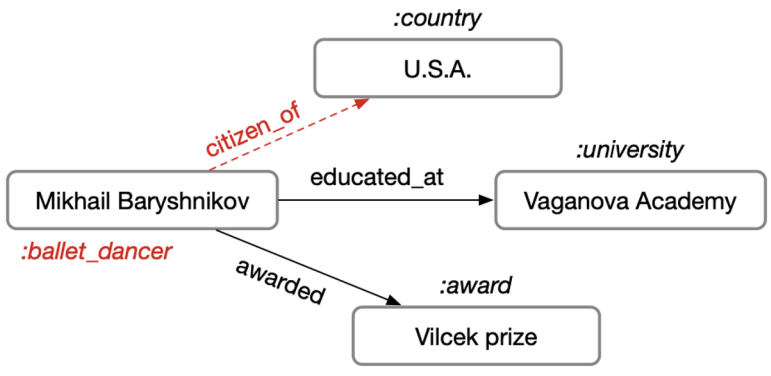
Knowledge Bases

- Knowledge graphs have various applications. (e.g. QA, IR)
- Incompleteness and lack of coverage hurts downstream applications.
- Statistical Relational Learning (SRL) => Mainly focusing on predicting missing info.
- A knowledge base fragment:
  - SRL: triples of the form (Subject, Predicate, Object)
  - Entities labeled with types (e.g., university, country)



Completion Tasks

- Entity Classification:
  - Assigning types or categorical properties to entities
- Link Prediction:
  - Recovery of missing triples
- Missing pieces of information are expected to reside within the encoded neighbourhood structure



Intro to R-GCN

- An encoder model is developed and applied to both tasks
  - An R-GCN model producing latent feature representations of entities
- Entity Classification task: Latent representations + Softmax classifier
- Link Prediction task: An autoencoder model consisting of ...
  - R-GCN as the encoder => Producing entity feature representation
  - A Tensor Factorization model as the decoder => Exploiting entity feature representation to predict edges

Background

Related Work – Neural Networks on Graphs

- Basic GNN: Scarselli et al. (2009)

Related Work – Neural Networks on Graphs

- Basic GNN: Scarselli et al. (2009)

- R-GCN Primary motivation: Idea of convolution on graphs and GCN

Related Work – Neural Networks on Graphs

- Basic GNN: Scarselli et al. (2009)
- R-GCN Primary motivation: Idea of convolution on graphs and GCN
- R-GCN as a sub-class of message passing neural networks (Gilmer et al. 2017):
  - Simplified representation: $$h_i^{(l+1)} = \sigma\left(\sum_{m \in \mathcal{M}_i} g_m(h_i^{(l)}, h_j^{(l)})\right)$$

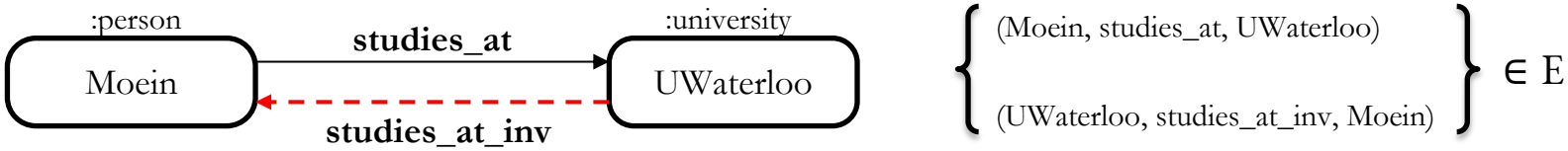
Related Work – Relational Modeling

- Tensor Factorization Model (DistMult)

Methodology

Graph Representation of Relational Data

- Directed and labeled multi-graph G = (V, E, R)
  - V: set of nodes (entities) => $v_i \in V$
  - R: set of relations => $r \in R$
    Contains relations in canonical direction (e.g. ***educated_at*** and ***educated_at_inv***)
  - E: set of labeled edges representing relations => $(v_i, r, v_j) \in E$



Proposed Method

$$h_i^{(l+1)} = \sigma\left(\sum_{m \in \mathcal{M}_i} g_m(h_i^{(l)}, h_j^{(l)})\right)$$

Simplified form of a MPNN

$$h_i^{(l+1)} = \sigma\left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)}\right)$$

R-GCN

$h_i^{(l)} \in \mathbb{R}^{d^{(l)}}$: Hidden state of node v_i in the l-th layer of NN

$\mathcal{M}_i$: Set of incoming messages for node v_i

$g_m(\cdot, \cdot)$: The form of incoming message to be accumulated and passed. It can be simply chosen as a linear transformation with a weight matrix W:

$$g_m(h_i, h_j) = W h_j$$

$\mathcal{N}_i^r$: The set of neighbour indices of node v_i under relation r

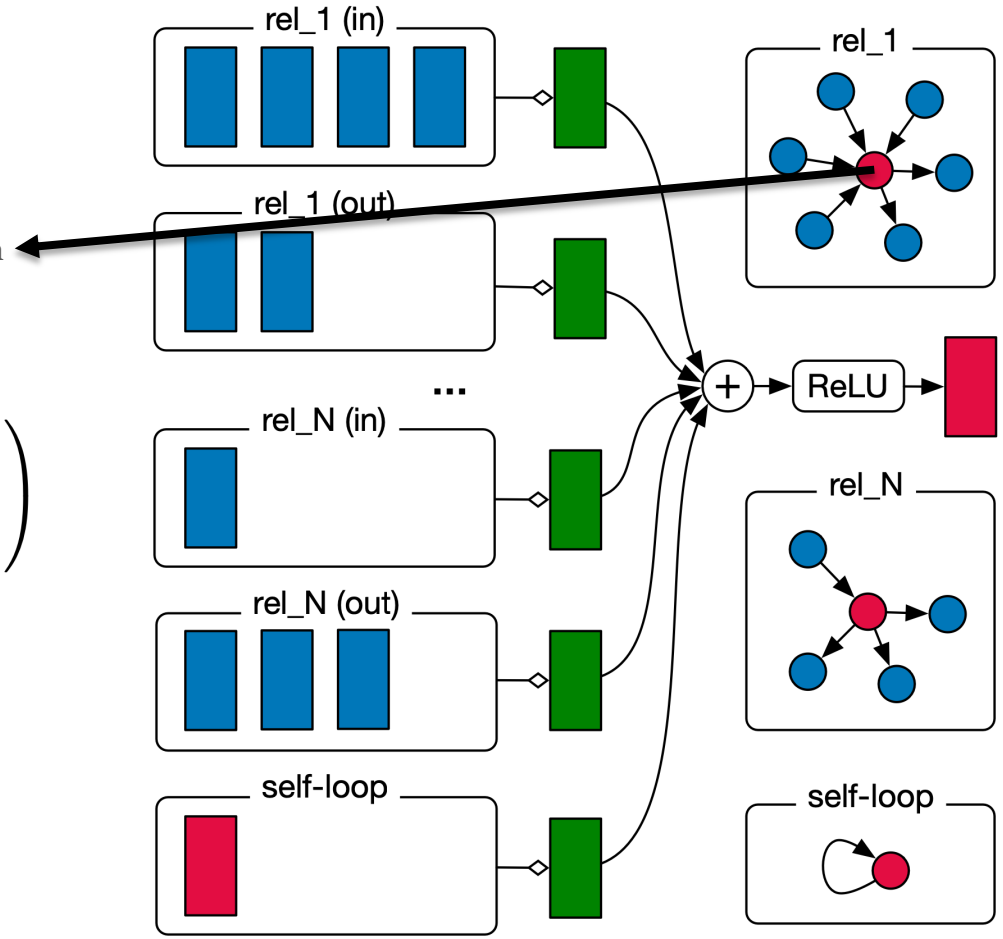
σ : Element-wise activation function such as ReLU

$c_{i,r}$: A normalization – can be chosen in advance:

$$c_{i,r} = |\mathcal{N}_i^r|$$

Proposed Architecture

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right)$$



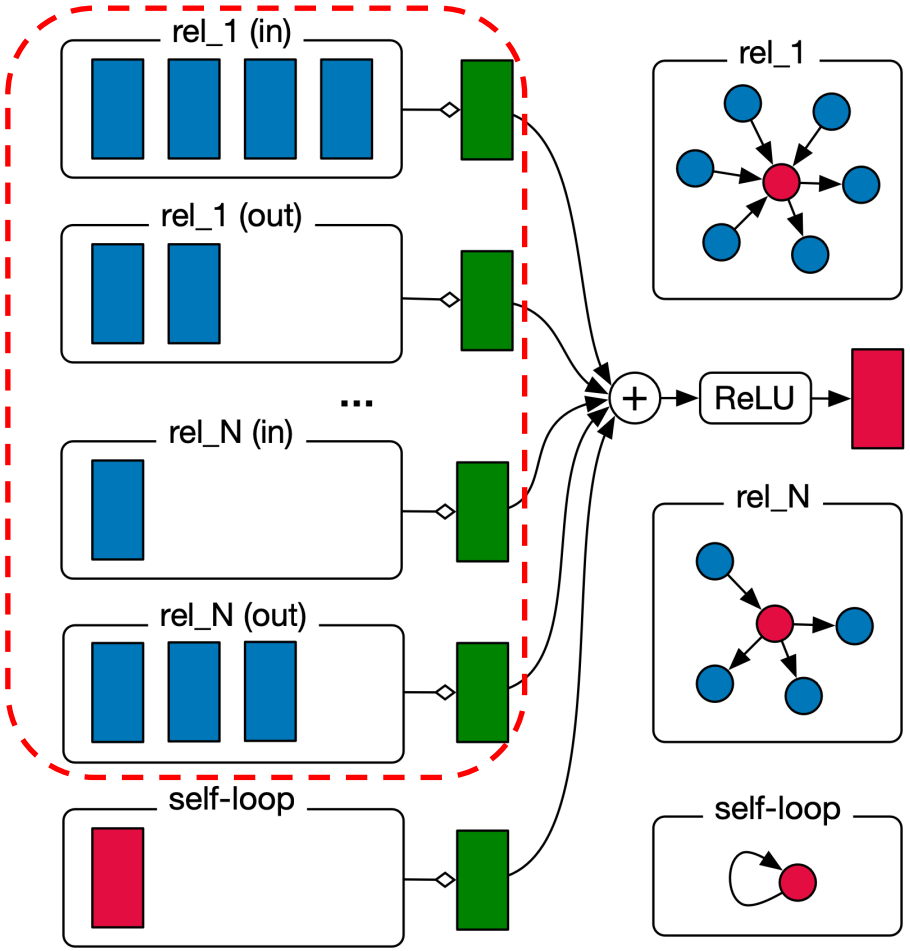
Proposed Architecture

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right)$$

$$\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)}$$

Accumulating transformed feature vectors of neighbours through a normalized sum

Using relation-specific transformation matrices (W_r)

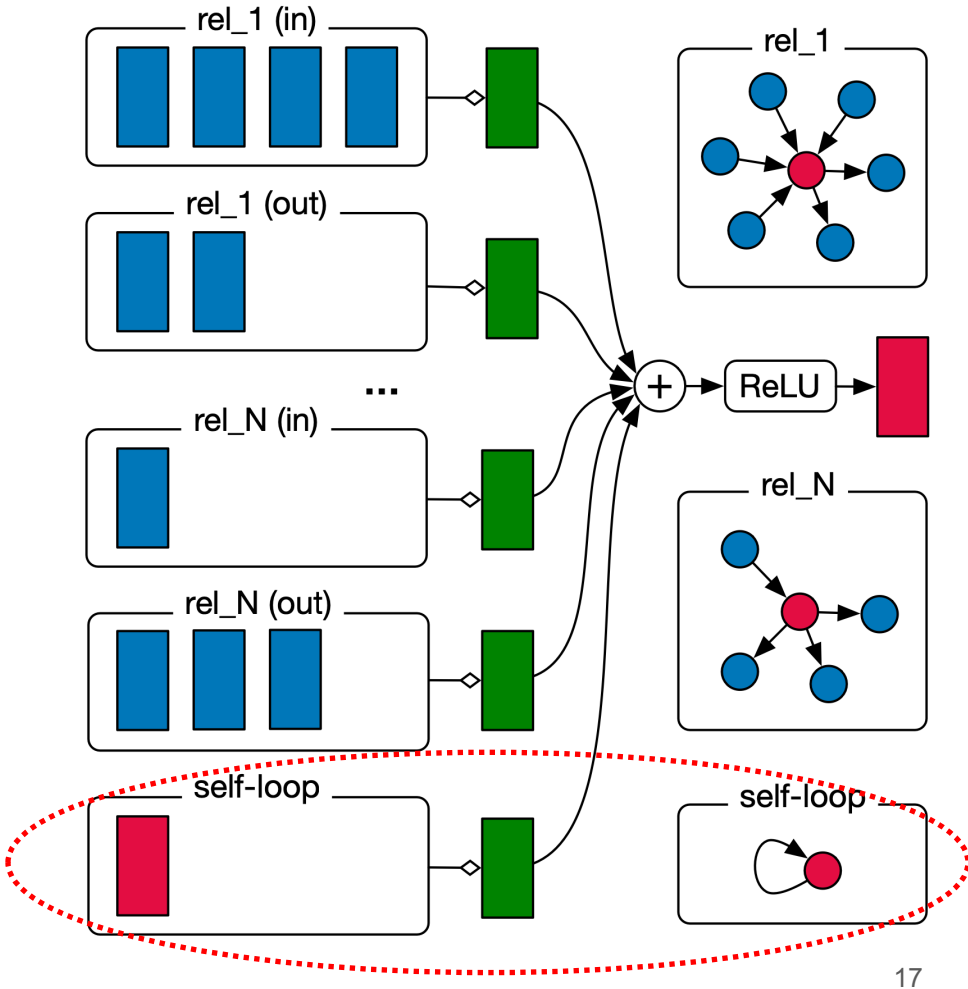


Proposed Architecture

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right)$$

$$W_0^{(l)}h_i^{(l)}$$

Adding a single self-connection of a special relation type to each node

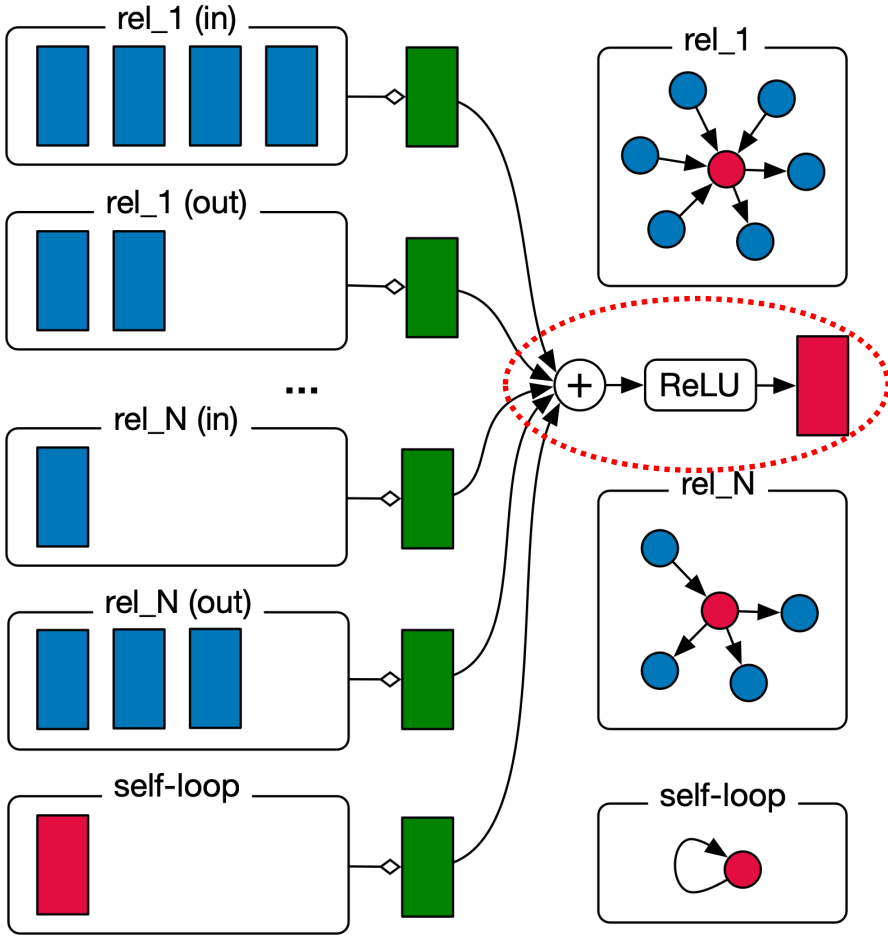


Proposed Architecture

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right)$$

Passing the summation through an activation function (e.g. ReLU)

Evaluating the value of $h_i^{(l+1)}$ **for every node** in the graph **in parallel**



Regularization

- Need for regularizing R-GCN layers' weights:
  - Rapid growth in **#parameters** with the increase in **#relations** in highly multi-relational data
  - Easily overfitting on rare relations and yielding models of very large size
- Regularization methods:
  - Basis-diagonal decomposition
  - Block-diagonal decomposition

Regularization

- Basis-diagonal decomposition
  - Each $W_r^{(l)}$ is defined as a linear combination of basis transformations
  - Effective weight sharing between different relation types to prevent model overfitting on rare relations.

$$W_r^{(l)} = \sum_{b=1}^{B} a_{rb}^{(l)} V_b^{(l)}$$

Regularization

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_n \end{bmatrix}$$

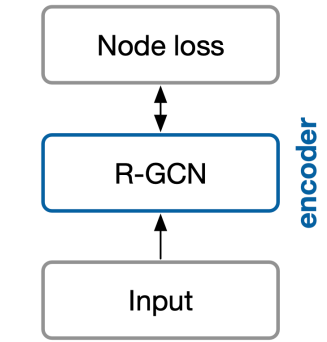
- Block-diagonal decomposition
  - Each $W_r^{(l)}$ is defined through the direct sum of a set of low-dimensional matrices
  - A sparsity constraint on the weight matrices for each relation type.

$$W_r^{(l)} = \bigoplus_{b=1}^{B} Q_{br}^{(l)} \quad \Rightarrow \quad W_r{}^{(l)} = diag\left(Q_{1r}{}^{(l)}, \ldots, Q_{Br}{}^{(l)}\right)$$

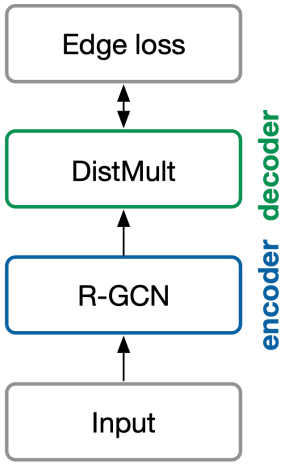
Overall R-GCN Structure

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right)$$

- Stacking L layers aligned to the definition of $h_i^{(l+1)}$
- First layer input: Unique one-hot vector for each node (featureless approach)
- Possible to incorporate pre-defined feature vectors for this class of models
- An overview of both models

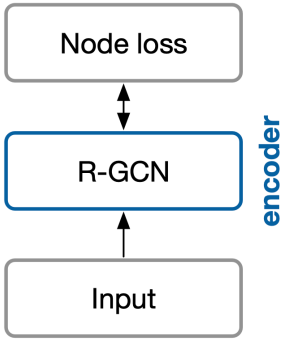


(a) Entity classification



(b) Link prediction

Entity Classification



- Semi-supervised node (entity) classification
- Minimizing Cross-Entropy loss on all labeled nodes (**ignoring unlabeled ones**):

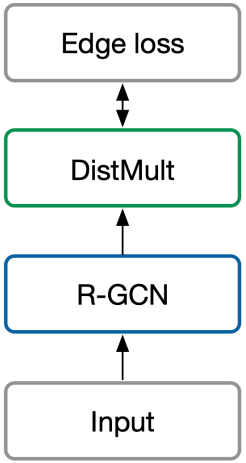
$\mathcal{Y}$: Set of nodes with labels

$h_{ik}^{(L)}$: The k-th entry of the network output for the i-th labeled node

t_{ik}: The respective ground truth label for that entry and node

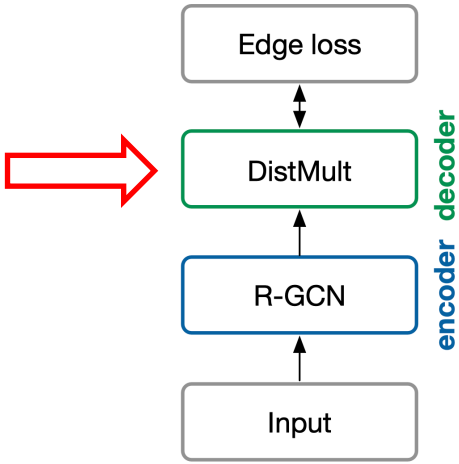
$$\mathcal{L} = -\sum_{i \in \mathcal{Y}} \sum_{k=1}^{K} t_{ik} \ln h_{ik}^{(L)}$$

Link Prediction



- Predicting new facts of the form of triples (subject, relation, object)
- Task: assign scores ($f(s,r,o)$) to possible edges (s,r,o)
  showing the likelihood of their existence.
- Method:
  - R-GCN entity encoder to map each entity $v_i \in \mathcal{V}$ to real-valued vectors. (Encoder)
  - DistMult: Scoring (s,r,o) triples using node representations through f: $\mathbb{R}^d \times \mathcal{R} \times \mathbb{R}^d \rightarrow \mathbb{R}$ (Decoder)
  - Key point: Relying on an encoder (R-GCN) by setting $e_i = h_i^{(L)}$

DistMult



- Known to perform well on standard link prediction benchmarks

- Score Calculation: $f(s, r, o) = e_s^T R_r e_o$

$$\mathbb{R}^d \times \mathcal{R} \times \mathbb{R}^d \rightarrow \mathbb{R}$$

Results

Entity Classification Experiments

- Four datasets in Resource Description Framework (RDF) format:
  - The value of a certain property is used as a classification target.
  - AIFB, MUTAG, BGS, and AM
  - The number of "Labeled" entities denotes the size of the subset of entities that have labels and are supposed to be classified.

Dataset	AIFB	MUTAG	BGS	AM
Entities	8,285	23,644	333,845	1,666,764
Relations	45	23	103	133
Edges	29,043	74,227	916,199	5,988,321
Labeled	176	340	146	1,000
Classes	4	2	2	11

Entity Classification Baselines

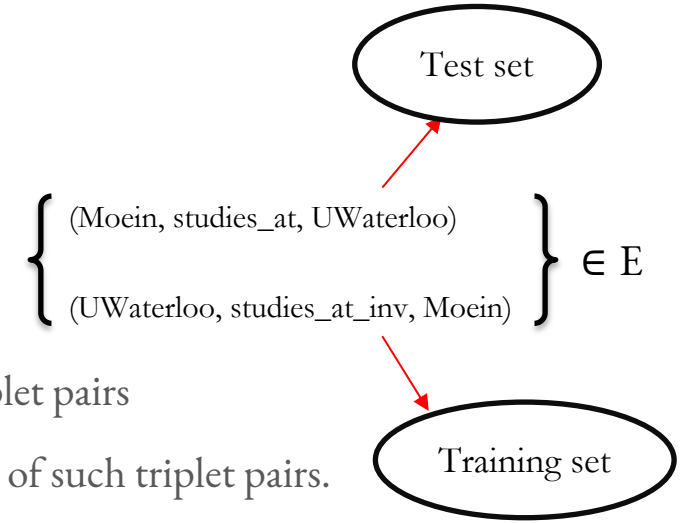
- Comparing against recent SOTA classification results:
  - RDF2Vec embeddings (RDF2Vec)
  - Weisfeller-Lehman kernels (WL)
  - Hand-designed feature extractors (Feat)

Entity Classification Experiments

- SOTA results in AIFB and AM dataset, and a gap in MUTAG and BGS
- Labeled entities in MUTAG and BGS only connected via high-degree hub nodes encoding a certain feature. => Choosing a fixed normalization constant might be problematic.

Model	AIFB	MUTAG	BGS	AM
Feat	55.55	77.94	72.41	66.66
WL	80.55	**80.88**	86.20	87.37
RDF2Vec	88.88	67.20	**87.24**	88.33
R-GCN	**95.83**	73.23	83.10	**89.29**

Link Prediction Experiments



- Experiment Datasets:
  - FB15k,WN18
  - A serious flaw in both datasets: presence of inverse triplet pairs
  - Reduces a large part of the task to a memorization task of such triplet pairs.
  - FB15k-237: As a result of removing such inverse triplets from FB15k
  - FB15k-237 used for primary evaluation, WN18 and FB15k still widely used

Link Prediction Datasets

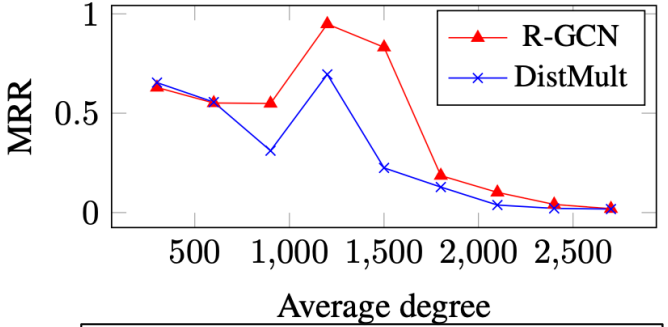
Dataset	WN18	FB15K	FB15k-237
Entities	40,943	14,951	14,541
Relations	18	1,345	237
Train edges	141,442	483,142	272,115
Val. edges	5,000	50,000	17,535
Test edges	5,000	59,071	20,466

Link Prediction Evaluation Metrics

- Mean Reciprocal Rank (MRR)
  - Rank-based metric
  - The average reciprocal ranking of the correct predictions
- Hits at n (H@n) => for n = 1, 3, 10
  - The fraction of true entities that appear in the first n entities of the sorted rank list
- Both bounded to (0, 1], the closer to one the better.

Link Prediction Experiments

- DistMult, ComplEx, LinkFeat, etc.



MRR for R-GCN and DistMult on the FB15k dataset as a function of avg. node (subject and object) degree

$$f(s,r,t)_{R-GCN+} = \alpha f(s,r,t)_{R-GCN} + (1-\alpha) f(s,r,t)_{DistMult}$$

	FB15k					WN18				
	MRR		Hits @			MRR		Hits @		
Model	Raw	Filtered	1	3	10	Raw	Filtered	1	3	10
LinkFeat		0.779			0.804		0.938			0.939
DistMult	0.248	0.634	0.522	0.718	0.814	0.526	0.813	0.701	0.921	0.943
R-GCN	0.251	0.651	0.541	0.736	0.825	0.553	0.814	0.686	0.928	0.955
R-GCN+	**0.262**	**0.696**	**0.601**	**0.760**	**0.842**	0.561	0.819	0.697	0.929	**0.964**
CP*	0.152	0.326	0.219	0.376	0.532	0.075	0.058	0.049	0.080	0.125
TransE*	0.221	0.380	0.231	0.472	0.641	0.335	0.454	0.089	0.823	0.934
HolE**	0.232	0.524	0.402	0.613	0.739	**0.616**	0.938	0.930	**0.945**	0.949
ComplEx*	0.242	0.692	0.599	0.759	0.840	0.587	**0.941**	**0.936**	**0.945**	0.947

Combining R-GCN with other scoring functions such as ComplEx?

Link Prediction Experiments

- Primary evaluation dataset (removed inverse relations):
  - LinkFeat fails to generalize to this dataset
  - R-GCN and R-GCN+ significantly outperforming DistMult as the main baseline method

	MRR		Hits @		
Model	Raw	Filtered	1	3	10
LinkFeat		0.063			0.079
DistMult	0.100	0.191	0.106	0.207	0.376
R-GCN	**0.158**	0.248	**0.153**	0.258	0.414
R-GCN+	0.156	**0.249**	0.151	**0.264**	**0.417**
CP	0.080	0.182	0.101	0.197	0.357
TransE	0.144	0.233	0.147	0.263	0.398
HolE	0.124	0.222	0.133	0.253	0.391
ComplEx	0.109	0.201	0.112	0.213	0.388

Conclusion

Conclusion

- R-GCN appears to be effective in two main SRL problems
- Entity Classification:
  - R-GCN showed competitive results as a trainable graph-based encoder.
- Link Prediction
  - The R-GCN + DistMult factorization model performed competitively on standard benchmarks.
  - This combination yielded a significant improvement (29.8%) on the main evaluation dataset (FB15k-237)

Future Work:

- Considering combination of R-GCN with other factorization models (e.g. ComplEx)
- Integrate entity features in R-GCN, benefiting both tasks

Thanks! ^_^

Any Questions?

Summary Questions

- What is the problem?
  - Failure of lots of downstream applications because of the incompleteness of large knowledge bases (or graphs)
- Why is it important?
  - KBs enabling multitude of applications by organizing factual knowledge
  - Importance to achieve maximum coverage to accomplish downstream tasks in the best way
- Why don’t previous methods work on this problem?
  - Their approach is mostly different and incomparable
  - R-GCN is expected to benefit from its GCN motivation and graph neighbourhood information.
- What is the solution to this problem?
  - Designing an auto-encoder model to accomplish two main tasks in SRL, benefiting from latent feature representations