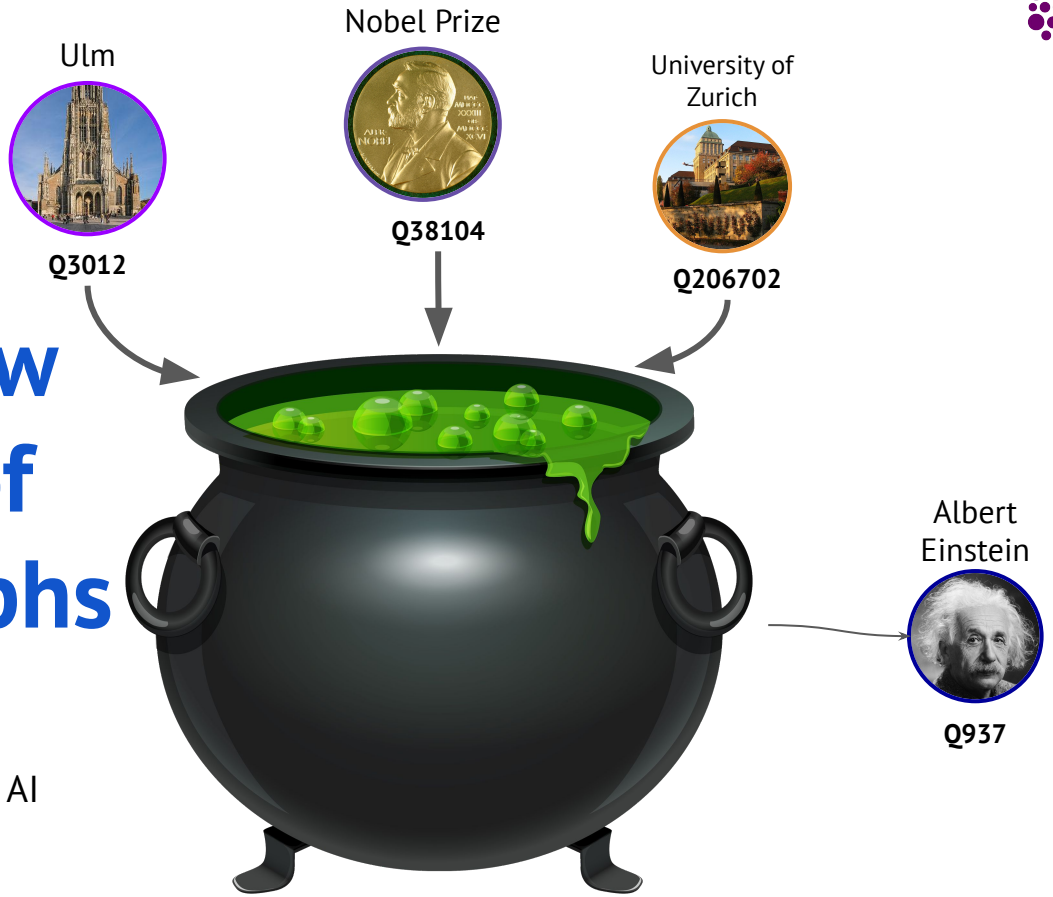
Beyond Shallow Embeddings of Knowledge Graphs

Michael Galkin
Postdoctoral Fellow @ Mila Quebec AI Institute & McGill University



The ImageNet Moment for KGs

Self-supervised pre-training → **Fine-tuning on a downstream task**

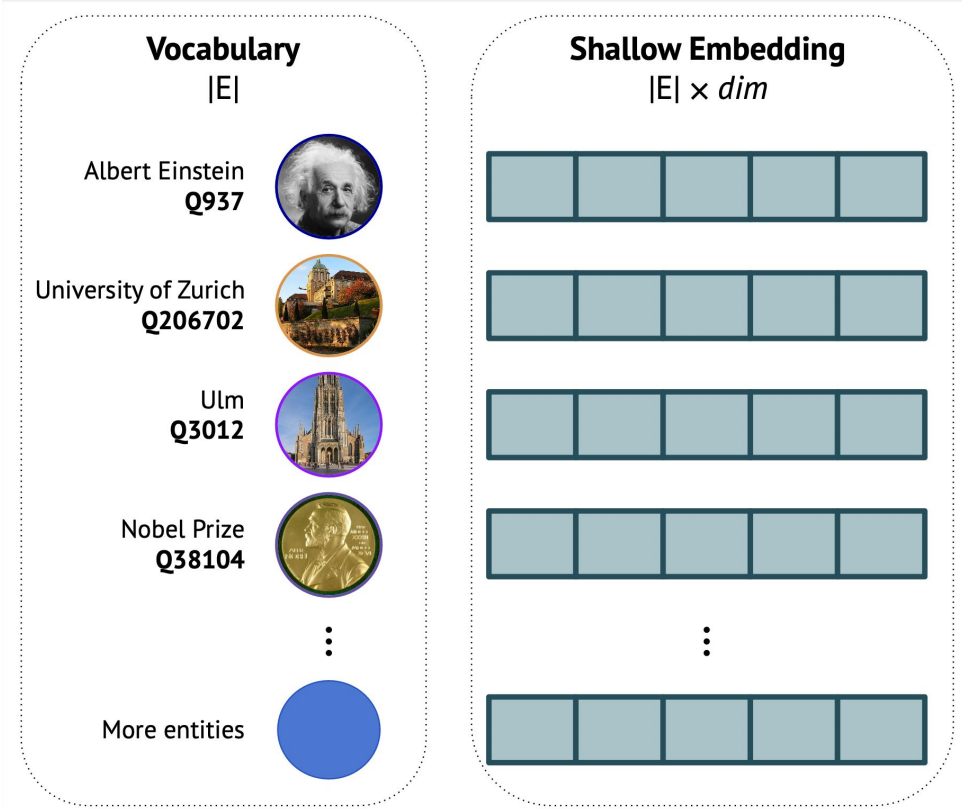
Wikidata: 100M nodes
Embs: [100M, dim] ?

PyTorch BigGraph

~200 GB

- Graph ML
- NLP
- Vision

Shallow Embedding



Transductive vs Inductive

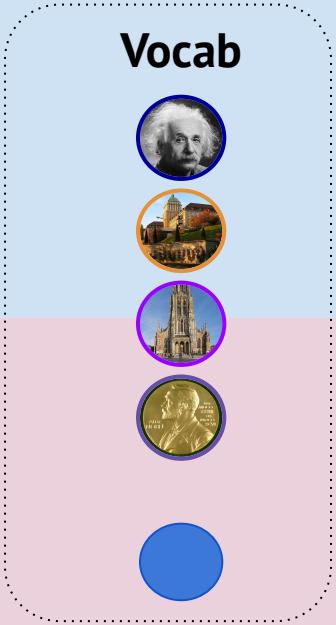
Shallow embeddings

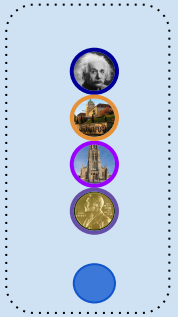
Transductive

Inductive

Training

Inference





New, unseen nodes (entities)

- Added to the seen graph
- Completely new inference graph

OGB WikiKG: Just 2.5M nodes

Leaderboard for ogbl-wikikg2

The MRR score on the test and validation sets. The higher, the better.

Package: >=1.2.4

Deprecated `ogbl-wikikg` leaderboard can be found here.

BERT-Large is ~340M params

Rank	Method	Test MRR	Validation MRR	Contact	References	#Params	Hardware	Date
1	**PairRE (200dim)**	0.5208 ± 0.0027	0.5423 ± 0.0020	Linlin Chao	Paper, Code	500,334,800	Tesla P100 (16GB GPU)	Jan 28, 2021
2	RotatE (250dim)	0.4332 ± 0.0025	0.4353 ± 0.0028	Hongyu Ren – OGB team	Paper, Code	1,250,435,750	Quadro RTX 8000 (45GB GPU)	Jan 23, 2021
3	TransE (500dim)	0.4256 ± 0.0030	0.4272 ± 0.0030	Hongyu Ren – OGB team	Paper, Code	1,250,569,500	Quadro RTX 8000 (45GB GPU)	Jan 23, 2021
4	ComplEx (250dim)	0.4027 ± 0.0027	0.3759 ± 0.0016	Hongyu Ren – OGB team	Paper, Code	1,250,569,500	Quadro RTX 8000 (45GB GPU)	Jan 23, 2021

BERT (340M params) - disruption in NLP ✅
KG embs (>1B params) - 🥺

Life beyond shallow embedding?

Do we really need to learn & store the whole shallow embedding matrix $|E| \times dim$

Trying to fit a 100M x 200 tensor on a Tesla V100 ->

Life beyond shallow embedding?

Neural Bellman-Ford Nets [1]

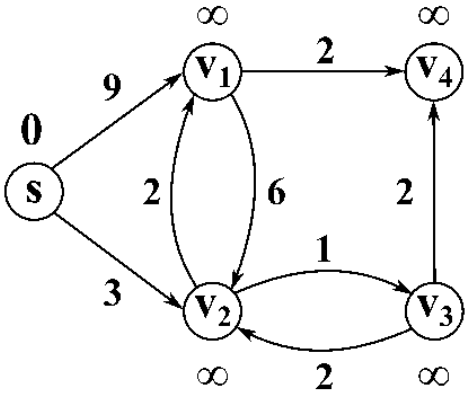
NodePiece [2]

[1] Zhu et al. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. NeurIPS'21
[2] Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations for Large Knowledge Graphs. arxiv:2021

Neural Bellman-Ford Nets

Vanilla Bellman-Ford for shortest paths

```
for v ∈ V
  do d[v] ← +∞
d[s] ← 0
for i ← 1 to |V| − 1
  do for (u, v) ∈ E
    if d[v] > d[u] + w(u, v)
      then d[v] ← d[u] + w(u, v)
return d
```



Source: Wikipedia

$$d[v] = min(d[u] + w(u, v), d[v])$$ <- vanilla Bellman-Ford iteration

[1] Zhu et al. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. NeurIPS'21

Generalized Bellman-Ford

$$d[v] = min(d[u] + w(u, v), d[v])$$

- Generalize + to any operator $\otimes$
- Generalize *min* to any operator $\oplus$
- $\oplus$ and $\otimes$ satisfy a **semiring** system

Boundary

$$\boldsymbol{h}_q^{(0)}(u, v) \leftarrow \mathbb{1}_q(u = v)$$

Generalized Bellman-Ford

$$\boldsymbol{h}_q^{(t)}(u, v) \leftarrow \left(\bigoplus_{(x,r,v) \in \mathcal{E}(v)} \boldsymbol{h}_q^{(t-1)}(u, x) \otimes \boldsymbol{w}_q(x, r, v) \right) \oplus \boldsymbol{h}_q^{(0)}(u, v)$$

[1] Zhu et al. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. NeurIPS'21

Parameterize -> Neural Bellman-Ford

$$\boldsymbol{h}_q^{(0)}(u,v) \leftarrow \mathbb{1}_q(u=v)$$

Boundary

Generalized Bellman-Ford

$$\boldsymbol{h}_q^{(t)}(u,v) \leftarrow \left(\bigoplus_{(x,r,v)\in\mathcal{E}(v)} \boldsymbol{h}_q^{(t-1)}(u,x) \otimes \boldsymbol{w}_q(x,r,v) \right) \oplus \boldsymbol{h}_q^{(0)}(u,v)$$

Parameterizing those operators we recover a relational GNN! 👀

$$\boldsymbol{h}_v^{(0)} \leftarrow \textsc{Indicator}(u,v,q)$$

$$\boldsymbol{h}_v^{(t)} \leftarrow \textsc{Aggregate}\left(\left\{ \textsc{Message}\left(\boldsymbol{h}_x^{(t-1)}, \boldsymbol{w}_q(x,r,v) \right) \middle| (x,r,v)\in\mathcal{E}(v) \right\} \cup \left\{ \boldsymbol{h}_v^{(0)} \right\} \right)$$

[1] Zhu et al. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. NeurIPS'21

NBF recovers classical graph algorithms

$$\boldsymbol{h}_v^{(0)} \leftarrow \textsc{Indicator}(u, v, q)$$

$$\boldsymbol{h}_v^{(t)} \leftarrow \textsc{Aggregate}\left(\left\{\textsc{Message}\left(\boldsymbol{h}_x^{(t-1)}, \boldsymbol{w}_q(x, r, v)\right) \middle| (x, r, v) \in \mathcal{E}(v)\right\} \cup \left\{\boldsymbol{h}_v^{(0)}\right\}\right)$$

Table 1: Comparison of operators in NBFNet and other methods from the view of path formulation.

Class	Method	MESSAGE $\boldsymbol{w}_q(e_i) \otimes \boldsymbol{w}_q(e_j)$	AGGREGATE $\boldsymbol{h}_q(P_i) \oplus \boldsymbol{h}_q(P_j)$	INDICATOR $⓪_q, ①_q$	Edge Representation $\boldsymbol{w}_q(e=(u,v))$
Traditional Link Prediction	Katz Index [25]	$\boldsymbol{w}_q(e_i) \times \boldsymbol{w}_q(e_j)$	$\boldsymbol{h}_q(P_i) + \boldsymbol{h}_q(P_j)$	$0, 1$	βw_e
	Personalized PageRank [37]	$\boldsymbol{w}_q(e_i) \times \boldsymbol{w}_q(e_j)$	$\boldsymbol{h}_q(P_i) + \boldsymbol{h}_q(P_j)$	$0, 1$	$\frac{\alpha w_{uv}}{\sum_{v' \in \mathcal{N}(u)} w_{uv'}}$
	Graph Distance [32]	$\boldsymbol{w}_q(e_i) + \boldsymbol{w}_q(e_j)$	$\min(\boldsymbol{h}_q(P_i), \boldsymbol{h}_q(P_j))$	$+\infty, 0$	w_e
Graph Theory Algorithms	Widest Path [3]	$\min(\boldsymbol{w}_q(e_i), \boldsymbol{w}_q(e_j))$	$\max(\boldsymbol{h}_q(P_i), \boldsymbol{h}_q(P_j))$	$-\infty, +\infty$	w_e
	Most Reliable Path [3]	$\boldsymbol{w}_q(e_i) \times \boldsymbol{w}_q(e_j)$	$\max(\boldsymbol{h}_q(P_i), \boldsymbol{h}_q(P_j))$	$0, 1$	w_e
Logic Rules	NeuralLP [63] / DRUM [41]	$\boldsymbol{w}_q(e_i) \times \boldsymbol{w}_q(e_j)$	$\boldsymbol{h}_q(P_i) + \boldsymbol{h}_q(P_j)$	$0, 1$	Weights learned by LSTM [21]
	NBFNet	Relational operators of knowledge graph embeddings [5, 62, 47]	Learned set aggregators [7]	Learned indicator functions	Learned relation embeddings

[1] Zhu et al. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. NeurIPS'21

NBFNet: No Trainable Entity Embeddings

$$\boldsymbol{h}_v^{(0)} \leftarrow \textsc{Indicator}(u, v, q)$$

- Initial node representations are initialized as a sum of adjacent relations
- Only one learnable bias (indicator) per node: *|E| x dim → |E| x 1*
- Only relations are trained -> **inductive out-of-the-box!**

$$\boldsymbol{h}_v^{(t)} \leftarrow \textsc{Aggregate}\left(\left\{\textsc{Message}\left(\boldsymbol{h}_x^{(t-1)}, \boldsymbol{w}_q(x, r, v)\right) \middle| (x, r, v) \in \mathcal{E}(v)\right\} \cup \left\{\boldsymbol{h}_v^{(0)}\right\}\right)$$

- Standard message passing relational GNN framework
- Message: any pairwise function (TransE, DistMult, RotatE, etc)
- Final node representation: *MLP([layer1, layer2, … layerK])*

[1] Zhu et al. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. NeurIPS'21

NBFNet: No Trainable Entity Embeddings

Table 8: Number of parameters in NBFNet. The number of parameters only grows with the number of relations $|\mathcal{R}|$, rather than the number of nodes $|\mathcal{V}|$ or edges $|\mathcal{E}|$. For FB15k-237 augmented with flipped triplets, $|\mathcal{R}|$ is 474. Our best configuration uses $T = 6$, $d = 32$ and hidden dimension $m = 128$. Note 2 NBFNet instances are required to model $p(v|u, q)$ and $p(u|v, q^{-1})$ respectively.

	#Parameter	
	Analytic Formula	**FB15k-237**
INDICATOR	$\|\mathcal{R}\|d$	15,168
MESSAGE	$\|\mathcal{R}\|Td$	91,008
AGGREGATE	$T(13d^2 + d)$	80,064
$f(\cdot)$	$m(d+1) + m + 1$	4,353
Total		190,593
2 NBFNet instances		381,186

SOTA RotatE
30M params

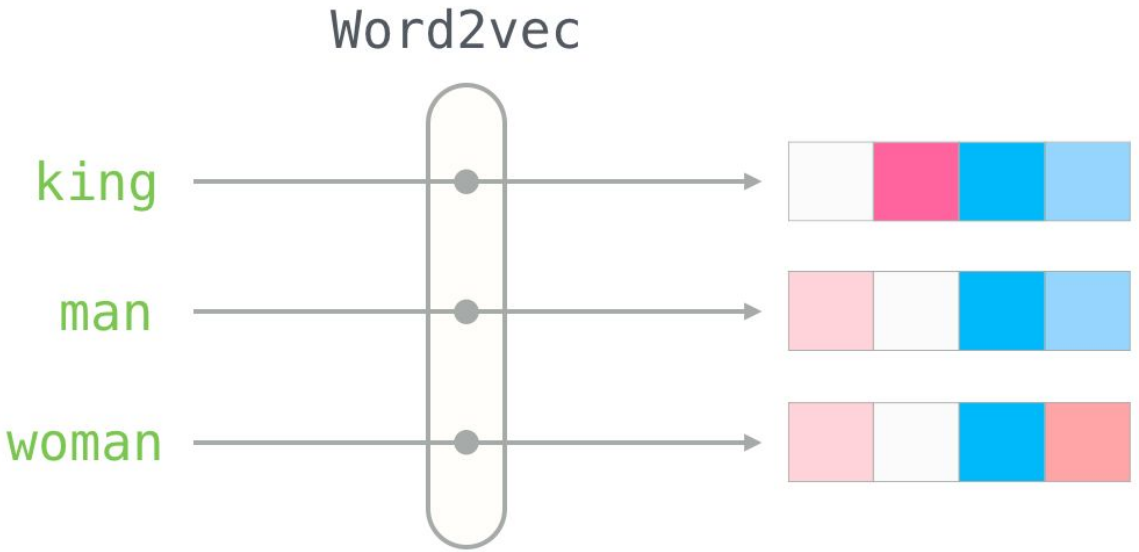
[1] Zhu et al. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. NeurIPS'21

New Transductive Link Prediction SOTA

Model	# Params	FB15k-237		WN18RR	
		MRR	H@10	MRR	H@10
RotatE	30 M	0.338	0.553	0.476	0.571
NBFNet	381 K	**0.378**	**0.563**	**0.547**	**0.661**

[1] Zhu et al. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. NeurIPS'21

Back to 2014



Unseen words = [OOV] (out-of-vocabulary)

Source: The Illustrated Word2vec by Jay Alammar

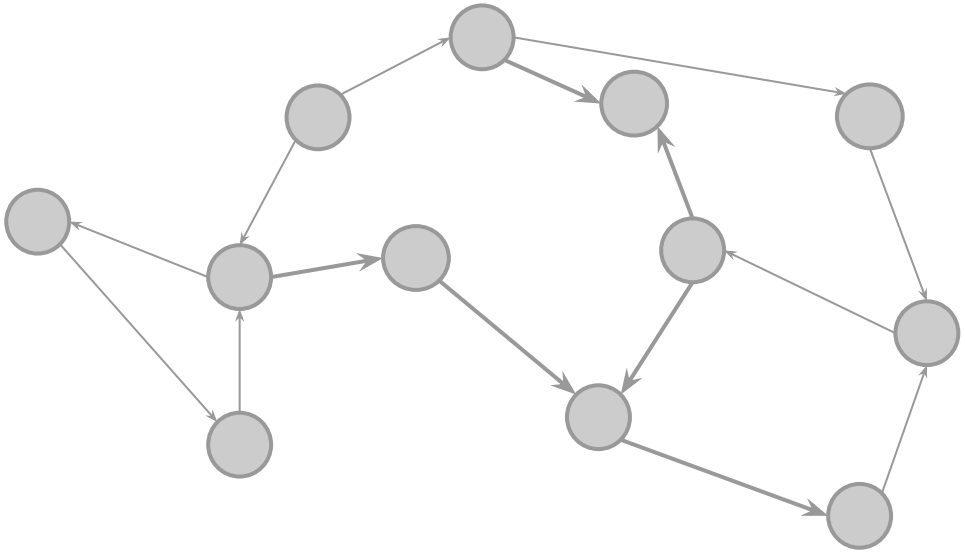
Byte-Pair Encoding / WordPiece

"I love tacos, apples, and tea!"

i	love	tacos	,	app	##les	,	and	t	##e	##a	!
6	7	8	5	10	11	5	9	30	41	37	3

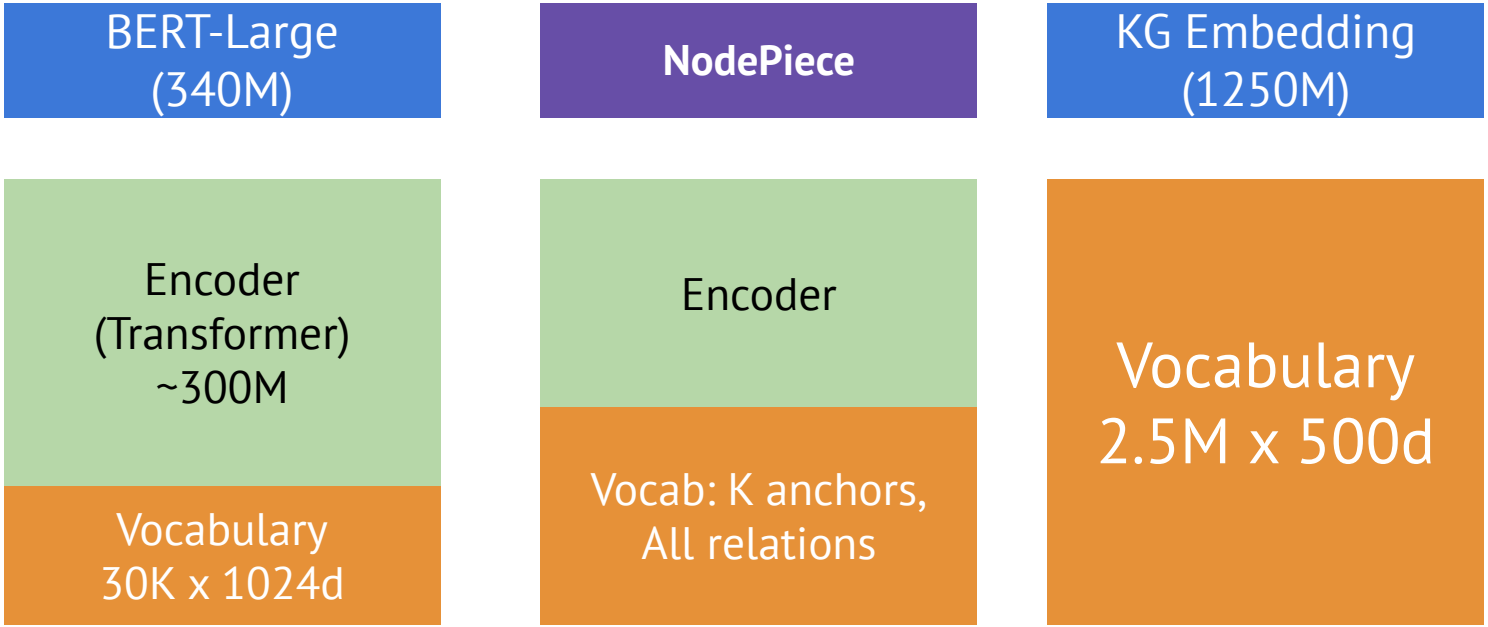
- Fixed-size vocab of subword units (30-50K)
- We can tokenize any unseen word

Tokenization + Graphs?



If nodes in a graph are
"**words**",
can we design a fixed-size
vocab of
"**sub-word**" units?

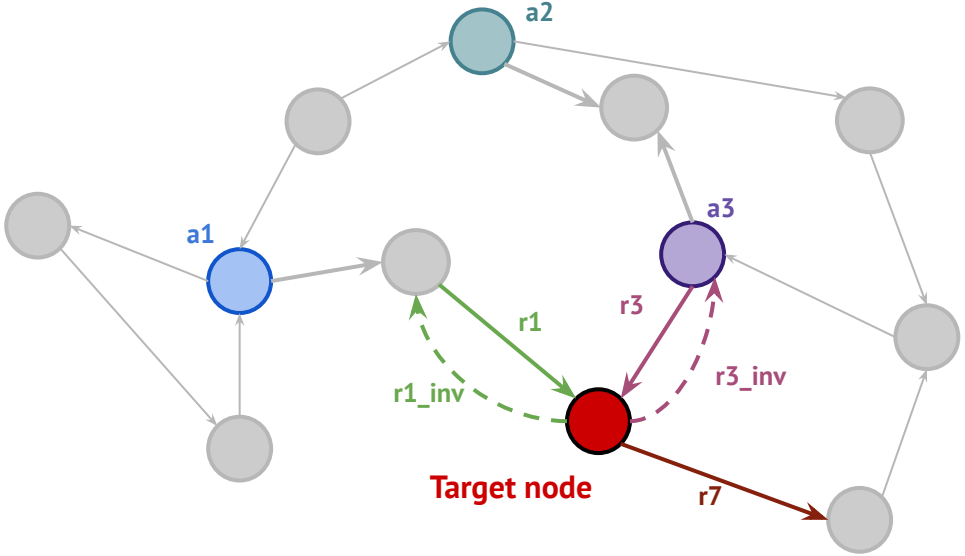
Tokenizing KGs

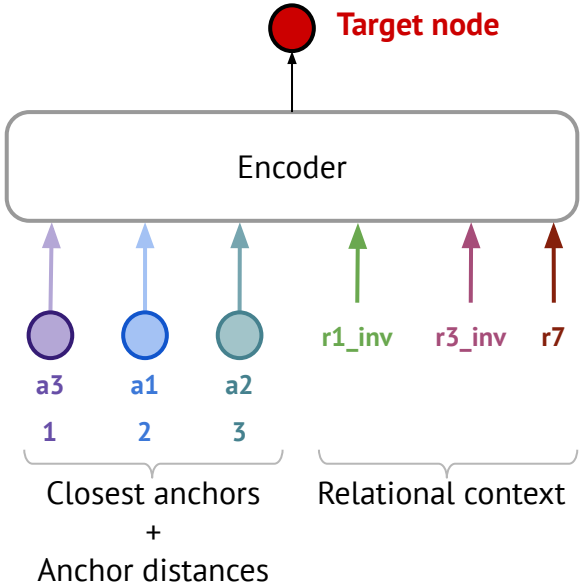


Tokenizing KGs

	Shallow embedding, only known words, otherwise OOV	Compositional representations, subword units
Language	Word2vec, GloVe	Byte-Pair Encoding, WordPiece
Graphs	All KG embedding algorithms (TransE, etc)	**NodePiece**

NodePiece - "*subword units*" for KGs



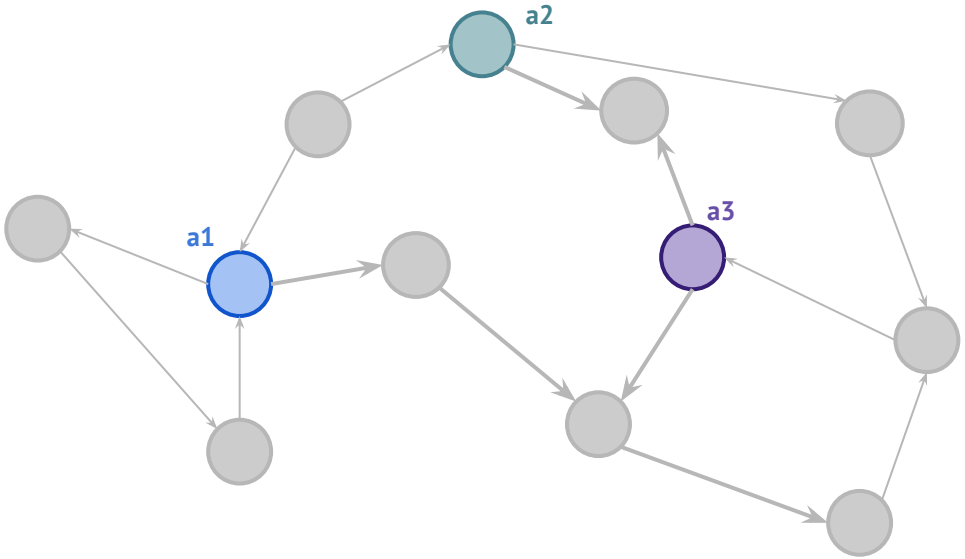


Vocabulary = Anchors + Relation types

Inductive out-of-the-box: unseen nodes are "tokenized" with the same Vocab

[1] Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations for Large Knowledge Graphs. arxiv:2021

Anchor Node Selection



Current strategy:
40% top degrees
40% top PageRank
20% random

[1] Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations for Large Knowledge Graphs. arxiv:2021

Tokenizing Einstein

Albert Einstein

$= f$ (

Ulm

,

Nobel Prize

,

Theoretical Physics

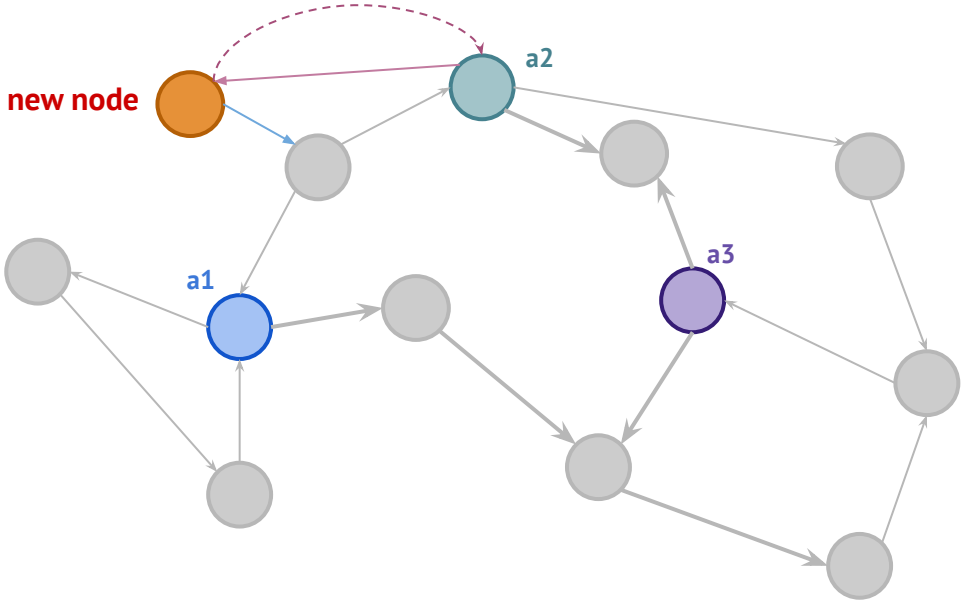
, place of birth **P19** , award received **P166** , occupation **P106** , academic degree **P512**)

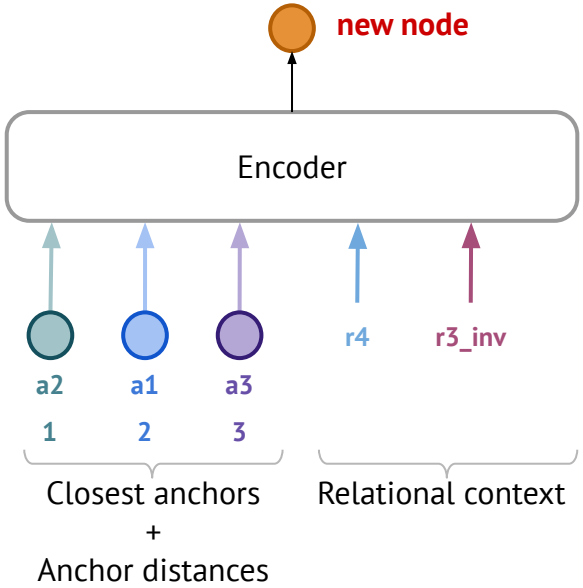
3 nearest anchors

4 unique outgoing relations in the context

[1] Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations for Large Knowledge Graphs. arxiv:2021

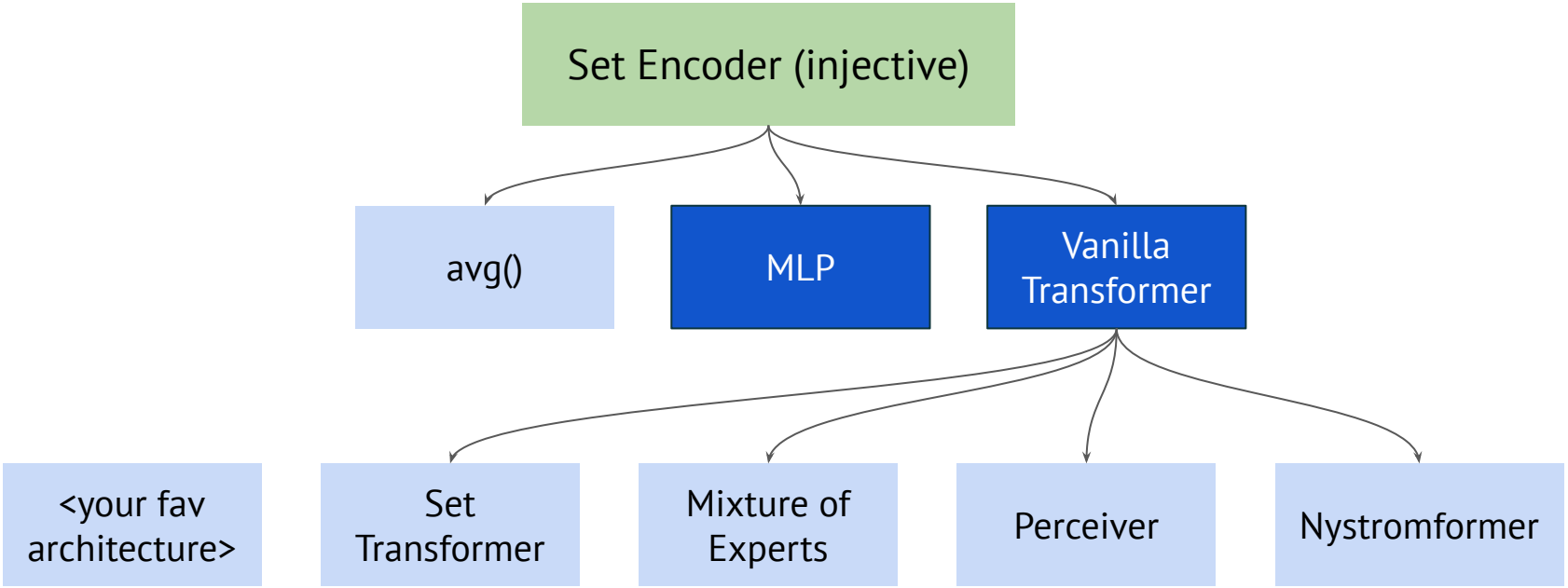
Unseen Node Tokenization





[1] Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations for Large Knowledge Graphs. arxiv:2021

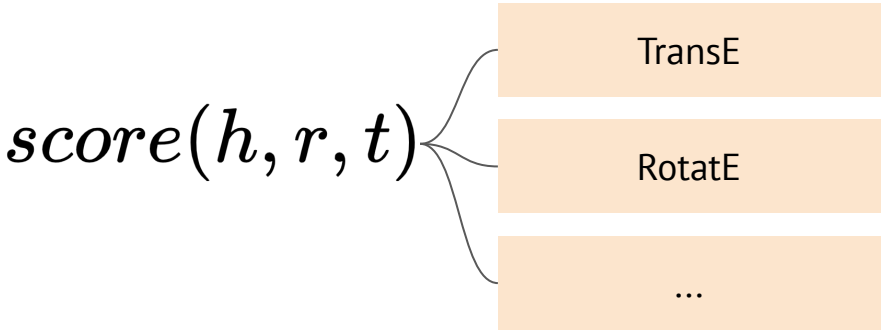
Set Encoder



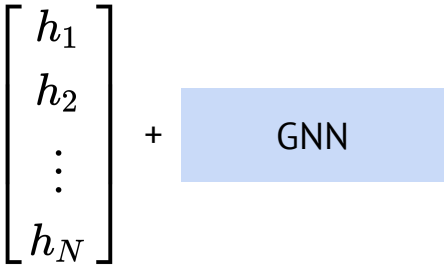
[1] Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations for Large Knowledge Graphs. arxiv:2021

Tasks

Link Prediction / Relation Prediction



Any GNN works, too!



[1] Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations for Large Knowledge Graphs. arxiv:2021

Transductive Link Prediction

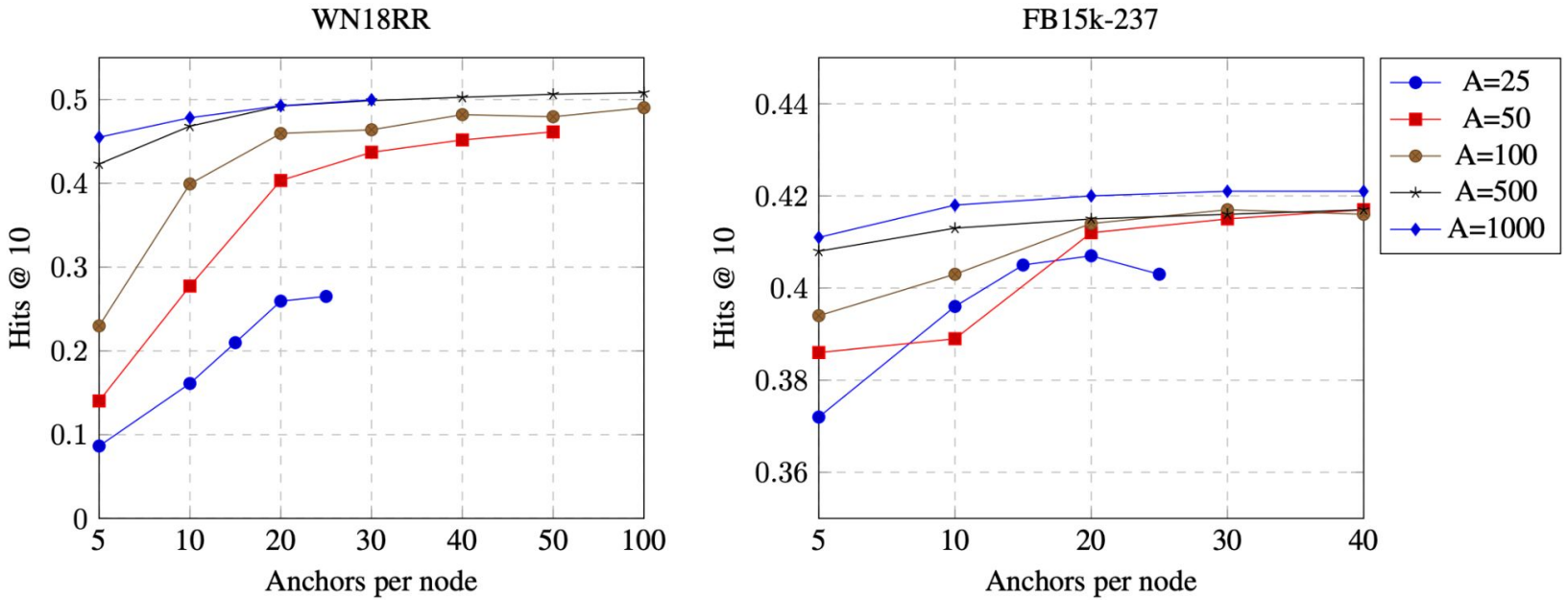
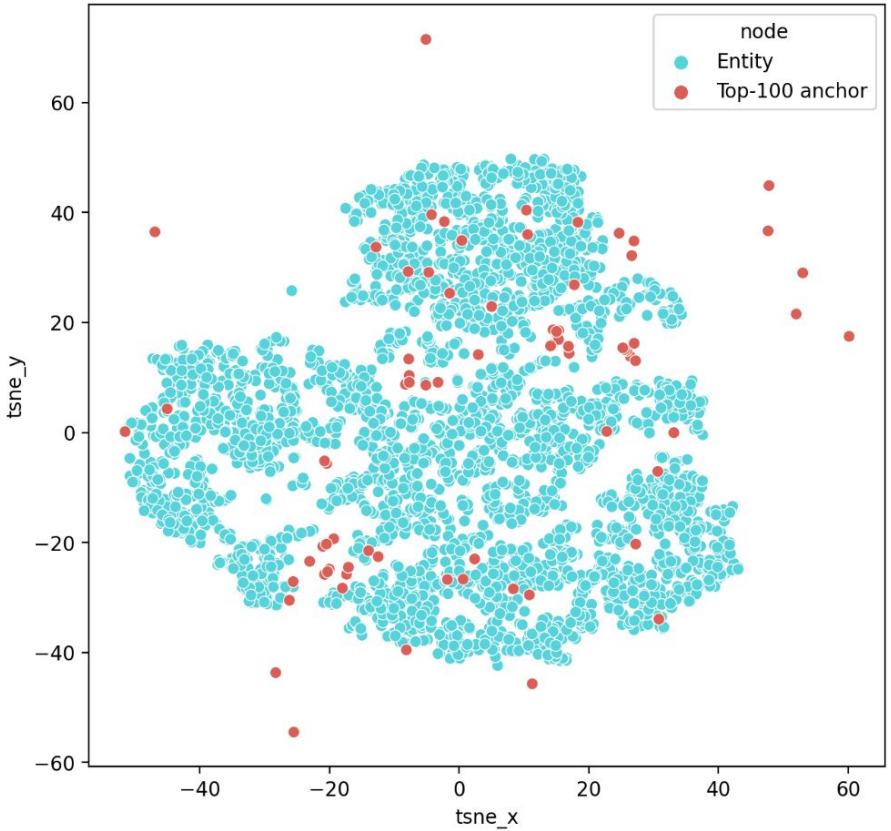
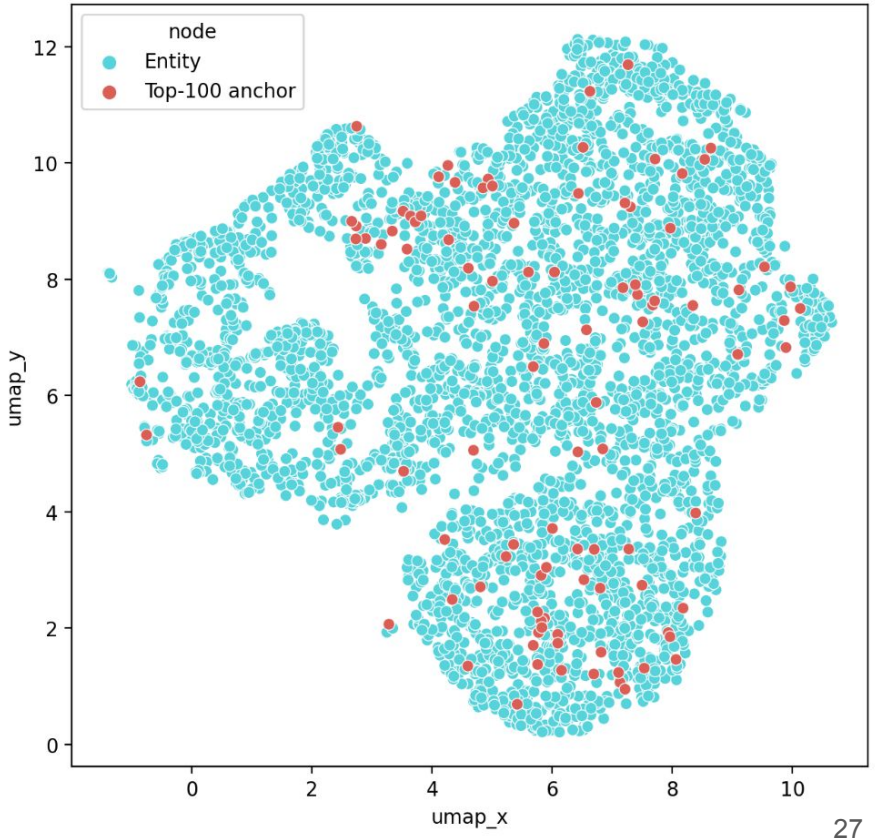


Figure 2: Combinations of total anchors A and anchors per node. Denser FB15k-237 saturates faster on smaller A while sparse WN18RR saturates at around 500 anchors.

[1] Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations for Large Knowledge Graphs. arxiv:2021

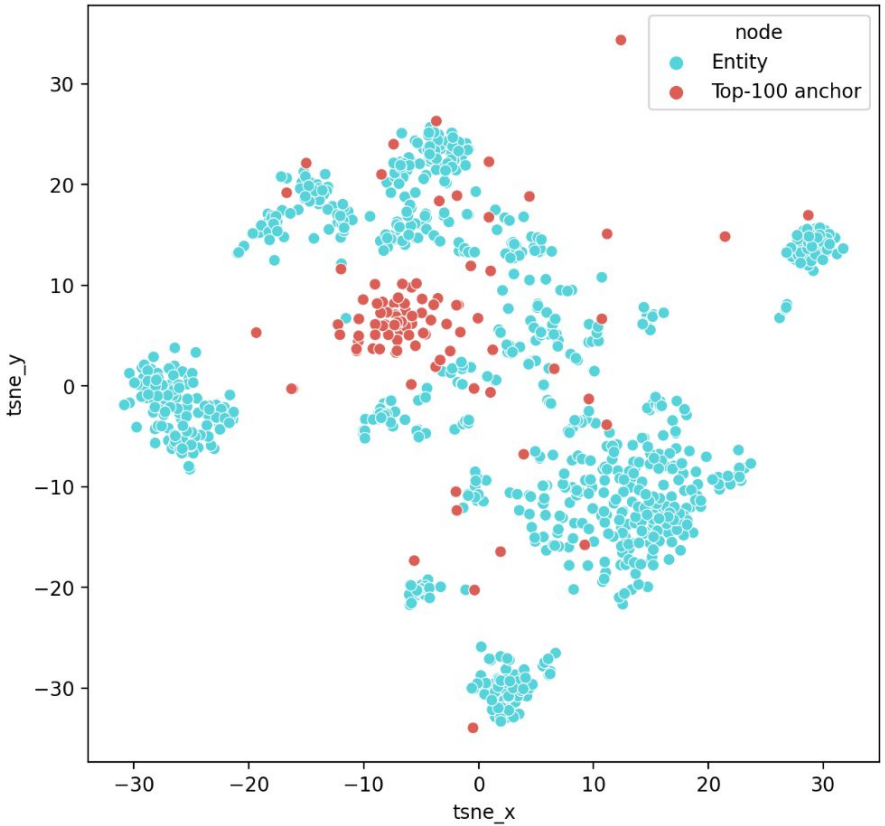
WN18RR anchors + entities

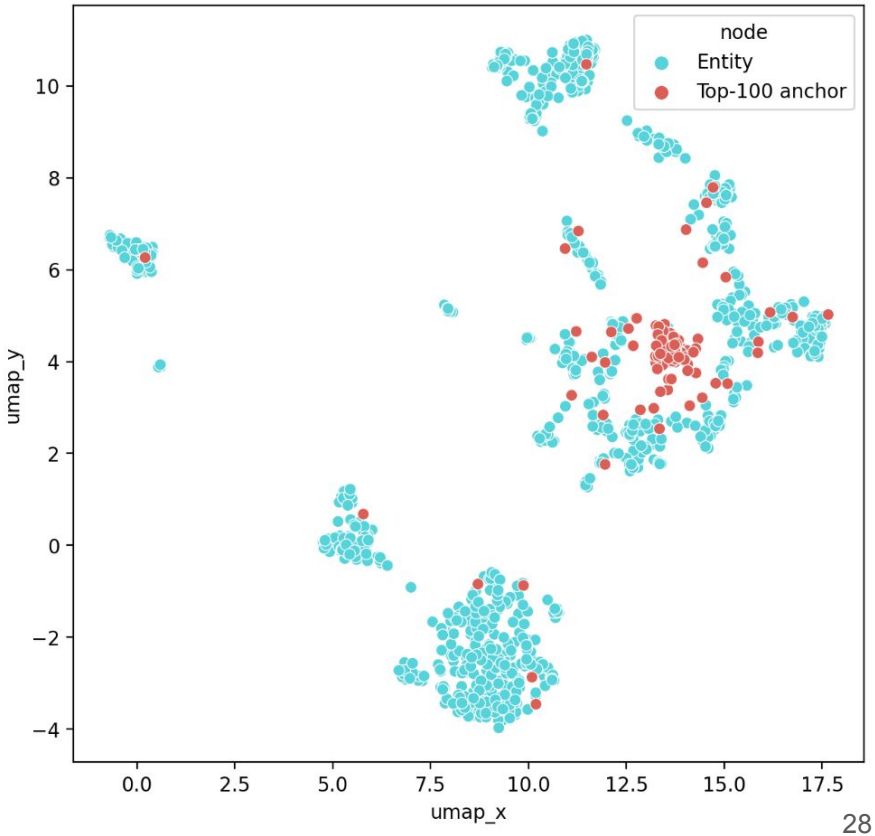




[1] Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations for Large Knowledge Graphs. arxiv:2021

FB15k-237 anchors + entities





[1] Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations for Large Knowledge Graphs. arxiv:2021

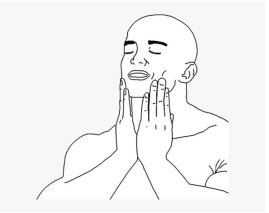
OGB WikiKG 2 : New SOTA

Leaderboard for ogbl-wikikg2

The MRR score on the test and validation sets. The higher, the better.

Package: >=1.2.4

Deprecated `ogbl-wikikg` leaderboard can be found here.



Rank	Method	Test MRR	Validation MRR	Contact	References	#Params	Hardware	Date
1	**NodePiece + AutoSF**	0.5703 ± 0.0035	0.5806 ± 0.0047	Mikhail Galkin (Mila)	Paper, Code	6,860,602	Tesla V100 (32 GB)	Jul 17, 2021
2	**AutoSF**	0.5458 ± 0.0052	0.5510 ± 0.0063	Yongqi Zhang (4Paradigm)	Paper, Code	500,227,800	Quadro RTX 8000 (45GB GPU)	Apr 2, 2021
3	**PairRE (200dim)**	0.5208 ± 0.0027	0.5423 ± 0.0020	Linlin Chao	Paper, Code	500,334,800	Tesla P100 (16GB GPU)	Jan 28, 2021
4	RotatE (250dim)	0.4332 ± 0.0025	0.4353 ± 0.0028	Hongyu Ren – OGB team	Paper, Code	1,250,435,750	Quadro RTX 8000 (45GB GPU)	Jan 23, 2021
5	TransE (500dim)	0.4256 ± 0.0030	0.4272 ± 0.0030	Hongyu Ren – OGB team	Paper, Code	1,250,569,500	Quadro RTX 8000 (45GB GPU)	Jan 23, 2021
6	ComplEx (250dim)	0.4027 ± 0.0027	0.3759 ± 0.0016	Hongyu Ren – OGB team	Paper, Code	1,250,569,500	Quadro RTX 8000 (45GB GPU)	Jan 23, 2021

~100x smaller

📜 > Neural Bellman-Ford Nets

< 📜

📜 > NodePiece < 📜

Contact  mikhail.galkin@mila.quebec

Socials  @michael_galkin