ICDE
MACAU 2019

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

4Paradigm 第四范式

北京郵電大學
Beijing University of Posts and Telecommunications

NSCaching: Simple and Efficient Negative Sampling for Knowledge Graph Embedding

Yongqi Zhang[❄], Quanming Yao[♈], Yingxia Shao[✦], Lei Chen[❄]

[❄]Hong Kong University of Science and Technology
[♈]4Paradigm Inc. (Beijing),
[✦]Beijing University of Posts and Telecommunications

[❄]{yzhangee, leichen}@cse.ust.hk, [♈]yaoquanming@4paradigm.com, [✦]shaoyx@bupt.edu.cn

Outline

- Introduction of KG Embedding
- Problems in Negative Sampling
- NSCaching Frameworks
- Experiments
- Summary

Knowledge Graph

Knowledge structure as graph

- Each node = an entity
- Each edge = a relation

Fact (triplet):

- (head, relation, tail)

Typical KGs:

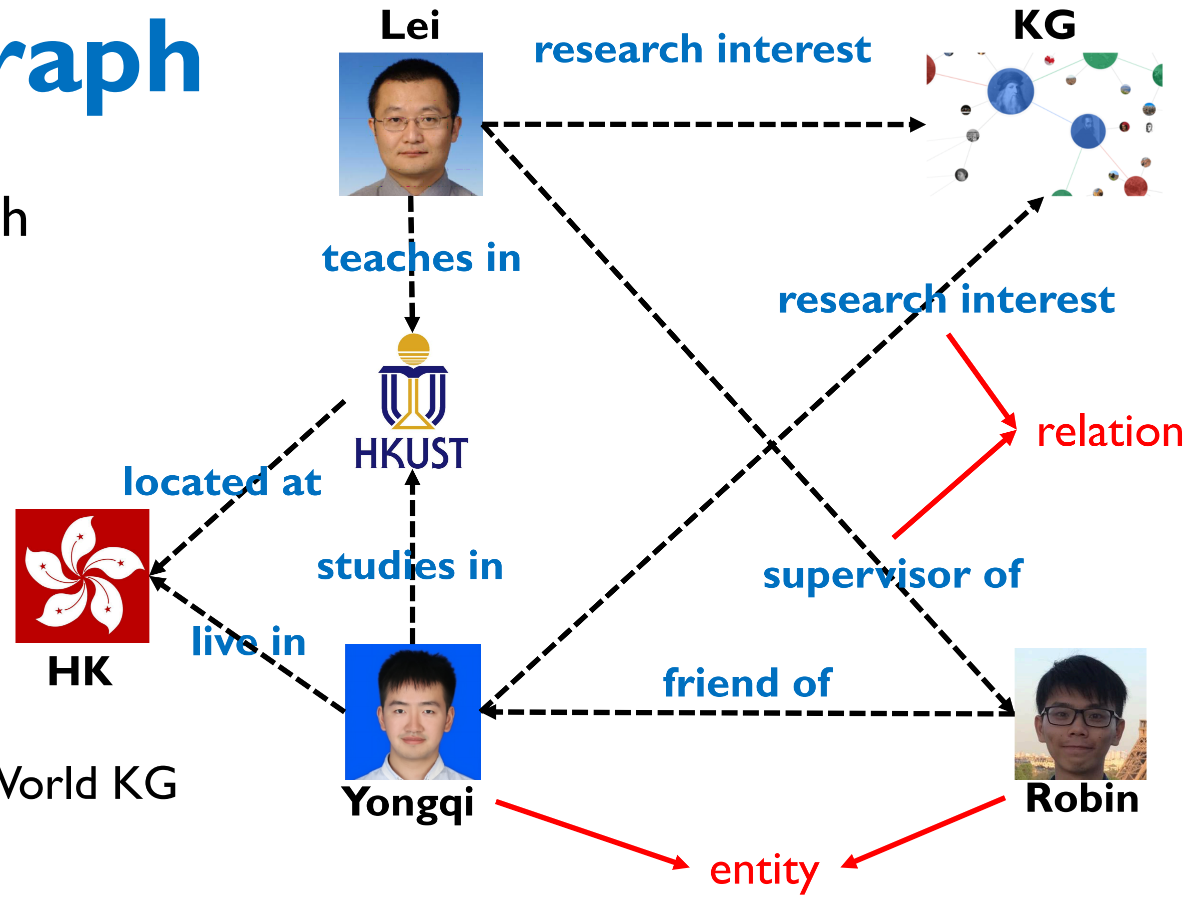
- WordNet: Linguistic KG
- Freebase, DBpedia, YAGO: World KG

Applications:

- Structured search [Dong et.al. KDD 2014]
- Question answering [Lukovnikov et.al. WWW 2017]
- Recommendation [Zhang et.al. KDD 2016]

KG completion

- (Lei, ?, Yongqi)



(Lei, teachers in, HKUST)
(HKUST, located in, HK)
(Yongqi, studies in, HKUST)
……
(head, relation, tail)
(h, r, t)

Why Embedding

Limitation to logical relations:

- Rules are restricted by manual design;
- Difficult to generalize to unseen entities/relations.

Computational Complexity

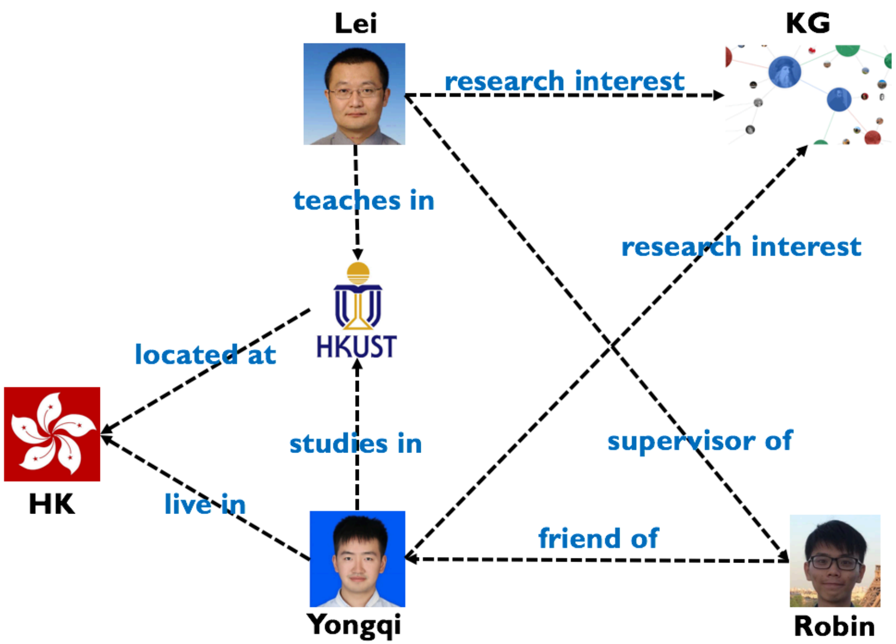
- Often NP-hard.
- Not trivial to parallelize, or use GPUs.

Embedding

- Learn from data
- Quick, generalizable

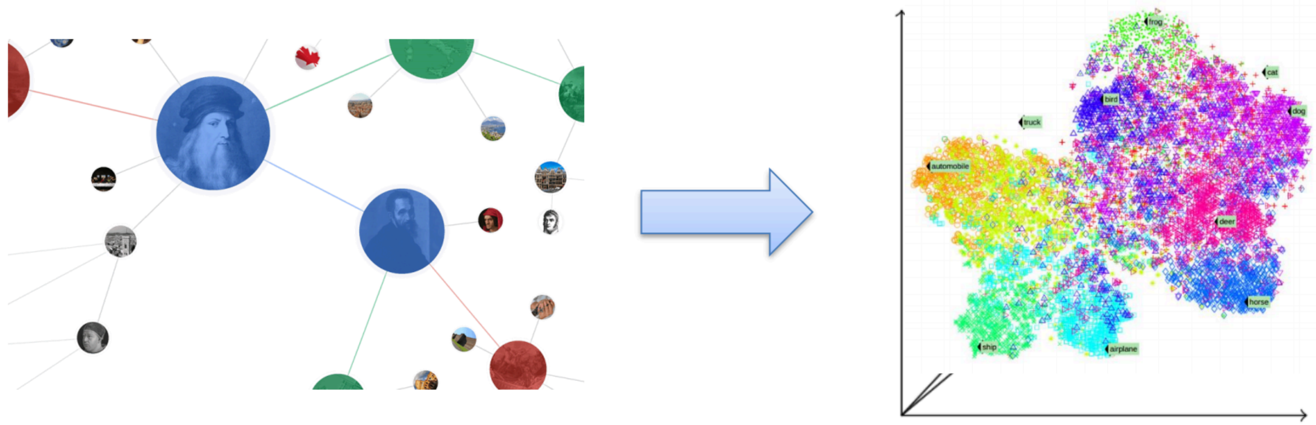
(Lei, ?, Yongqi)

(Yongqi, equals, Yong-QI) ?



Knowledge Graph Embedding

Encode KGs $\mathcal{G} = (\mathcal{E}, \mathcal{R})$ into low-dimensional vector spaces $\mathbb{R}^{d_1}$ and $\mathbb{R}^{d_2}$, while capturing nodes' and edges' connection properties.



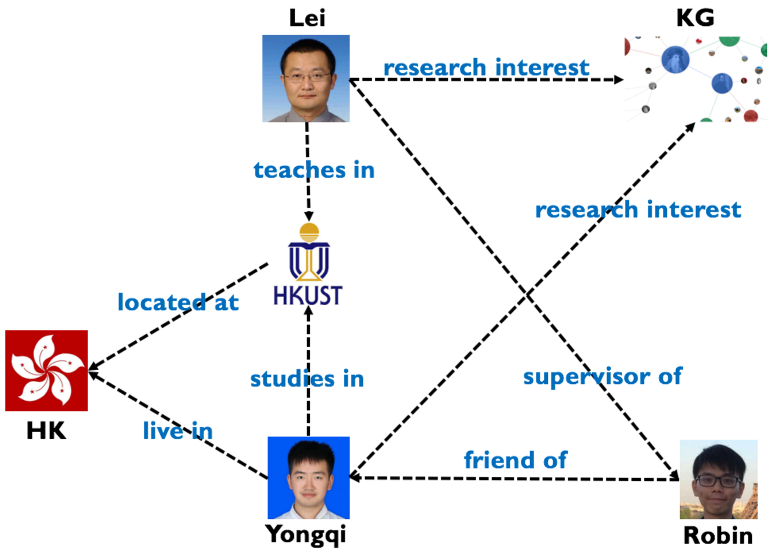
A scoring function $f(h, r, t)$ is designed to capture the interactions (similarity) between two entities based on a relation by their embeddings.

TransE:	$f(h,r,t) = -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_1$
DistMult:	$f(h,r,t) = <\mathbf{h}, \mathbf{r}, \mathbf{t}>$

Target, learn embeddings so that:

- f is maximum on a set of positive triplets $\mathcal{S} = \{(h, r, t)\}$;
- f is minimum on a set of negative triplets $\bar{\mathcal{S}} = \{(\bar{h}, r, \bar{t})\}$.

Negative Sampling



A KG only contains observed facts (positive triplets);

Non-observed ones are assumed to be negative with large probability.

Positive	Negative
(Lei, *teaches in*, HKUST)	(Lei, *teaches in*, Robin), (HK, *teaches in*, HKUST), (Lei, *research interest*, HKUST)
(HKUST, *located at*, HK)	(HKUST, *located at*, KG), (Yongqi, *located at*, HK), (HKUST, *friend of*, HK)

Given a positive triplet (h, r, t), the set of negative triplets is

$$\bar{\mathcal{S}}_{(h,r,t)} = \{(\bar{h}, r, t) \notin \mathcal{S} | \bar{h} \in \mathcal{E}\} \cup \{(h, r, \bar{t}) \notin \mathcal{S} | \bar{t} \in \mathcal{E}\},$$

$\{(h, \bar{r}, t) \notin \mathcal{S} | \bar{r} \in \mathcal{E}\}$ is not included since it is more likely to be false negative.

[wang et. al. TKDE]

General Framework of KG Embedding

Algorithm 1 General framework of KG embedding.

Input: training set $\mathcal{S} = \{(h, r, t)\}$, embedding dimension d and scoring function f;

initialize the embeddings for each $e \in \mathcal{E}$ and $r \in \mathcal{R}$.
for $i = 1, \cdots, T$ **do**
    sample a mini-batch $\mathcal{S}_{\text{batch}} \in \mathcal{S}$ of size m;
    for each $(h, r, t) \in \mathcal{S}_{\text{batch}}$ **do**
        sample a negative triplet $(\bar{h}, r, \bar{t}) \in \bar{\mathcal{S}}_{(h,r,t)}$;
        // negative sampling
        update parameters of embeddings w.r.t. the gradients using (i). translational distance models:

$$\nabla \left[\gamma - f(h, r, t) + f\left(\bar{h}, r, \bar{t}\right)\right]_{+}, \tag{3}$$

        or (ii). semantic matching models:

$$\nabla \ell\left(+1, f(h, r, t)\right) + \nabla \ell\left(-1, f\left(\bar{h}, r, \bar{t}\right)\right); \tag{4}$$

    end for
end for

positive

(Lei, *teaches in*, HKUST)
(HKUST, *located at*, HK)

negative

(Lei, *teaches in*, Robin)
(Yongqi, *located at*, HK)

$$\bar{\mathcal{S}}_{(h,r,t)} = \left\{\left(\bar{h}, r, t\right) \notin \mathcal{S} \,|\, \bar{h} \in \mathcal{E}\right\} \cup \left\{\left(h, r, \bar{t}\right) \notin \mathcal{S} \,|\, \bar{t} \in \mathcal{E}\right\}$$

Outline

- Introduction of KG Embedding
- Problems in Negative Sampling
- NSCaching Frameworks
- Experiments
- Summary

Problems in Negative Sampling

Low quality negative samples become less informative gradually

- Positive: *(Lei, teaches in, HKUST)*
- Low-quality: *(Lei, teaches in, orange)*
- High-quality: *(Lei, teaches in, HKU)*

The quality of negative samples matters!

$$\bar{\mathcal{S}}_{(h,r,t)} = \{(\bar{h},r,t) \notin \mathcal{S} | \bar{h} \in \mathcal{E}\} \cup \{(h,r,\bar{t}) \notin \mathcal{S} | \bar{t} \in \mathcal{E}\}$$

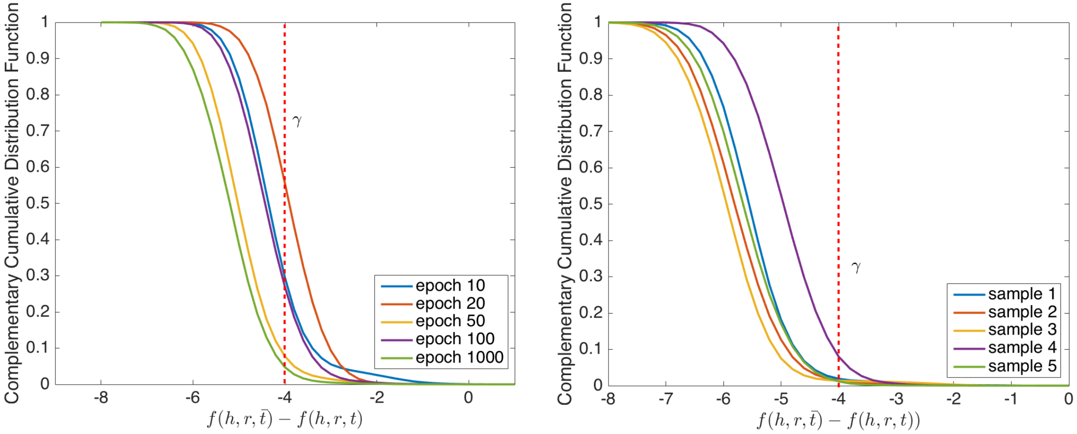
Observations:

Scores can be used to measure the qualities, but the score distribution of negative triplets is highly skewed.

Dynamic

Rare

Complex

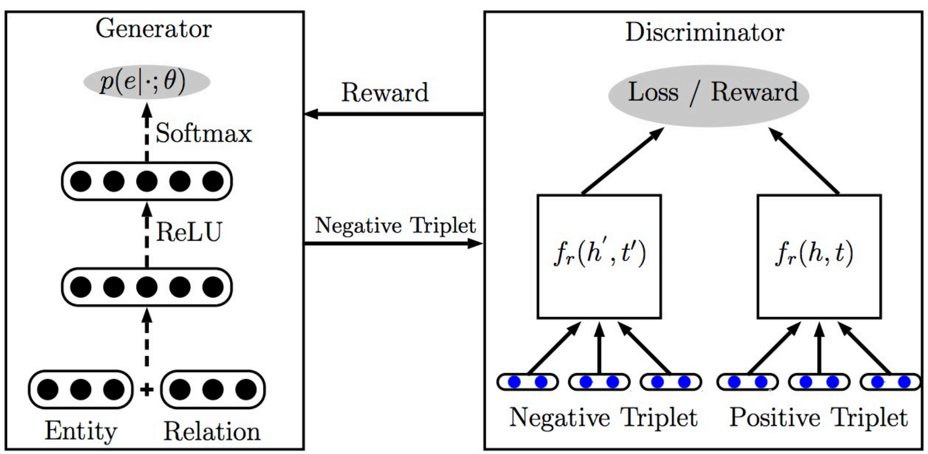


Challenges

How to model the dynamic distribution of negative triplets?

How to sample high-quality negative triplets in an efficient way?

GAN-base method ---- learn a generator as the sampler



[Wang et.al. AAAI 2018]

- Needs to learn an extra model.
- Sampling is not efficient.
- Training suffers from instability.

Motivations

Since the KG embedding itself contains information about triplets quality, we can

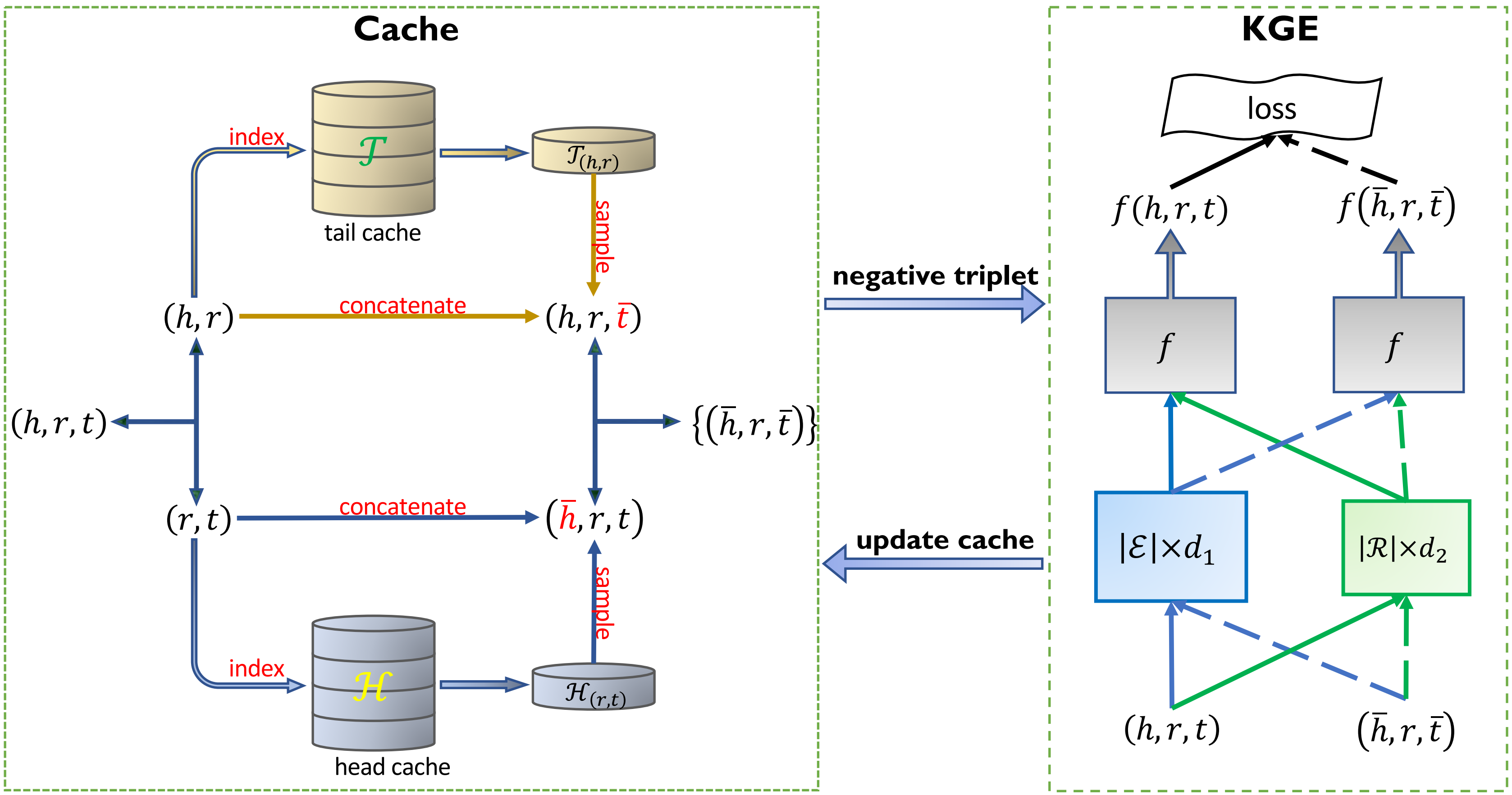
- Use a small amount of extra memory, which caches negative samples with large scores during training;
- Keep the cache updating periodically;
- Sample the negative triplets directly from the cache.

Dynamic Efficiency

Outline

- Introduction of KG Embedding
- Problems in Negative Sampling
- NSCaching Frameworks
- Experiments
- Summary

Overview of NSCaching



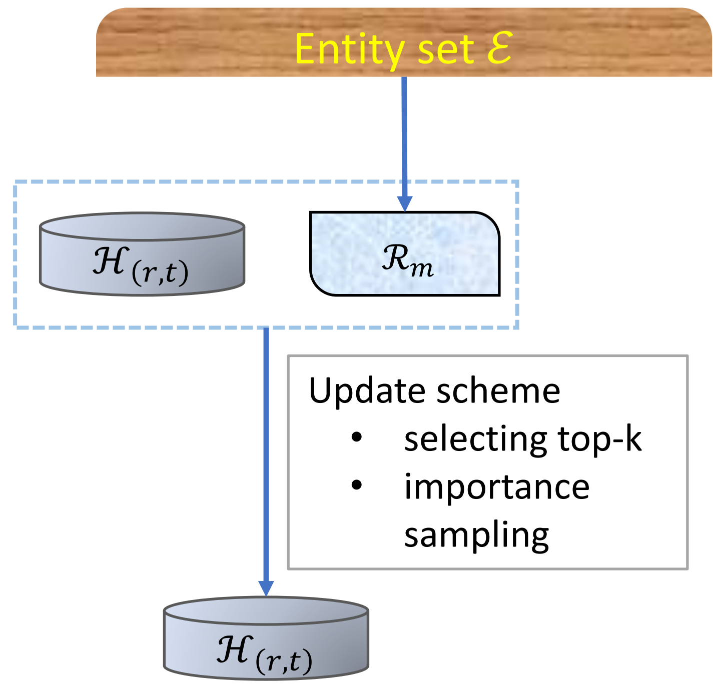
$$\bar{\mathcal{S}}_{(h,r,t)} = \{(\bar{h},r,t) \notin \mathcal{S} | \bar{h} \in \mathcal{E}\} \cup \{(h,r,\bar{t}) \notin \mathcal{S} | \bar{t} \in \mathcal{E}\}$$

Core steps

Dynamic
Efficiency
Effectiveness

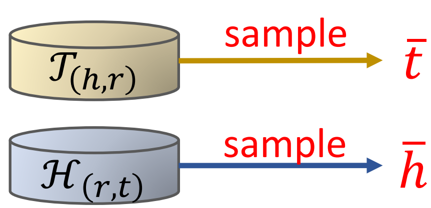
Updating the cache

- Randomly sample a small subset from all the candidates
- Selecting high-quality negative samples to update the cache



Sampling from the cache

- Uniform sampling
- Importance sampling
- Selecting the top



It can also benefit from self-paces learning!

Complexity

	strategy		minibatch computation		model
	negative sample	training	time	space	parameters
baseline	uniform random	gradient descent (from scratch)	$O(md)$	$O(md)$	$(\lvert\mathcal{E}\rvert + \lvert\mathcal{R}\rvert)d$
IGAN [33]	GAN	reinforce learning (with pretrain)	$O(m\lvert\mathcal{E}\rvert d)$	$O(m\lvert\mathcal{E}\rvert d)$	$3(\lvert\mathcal{E}\rvert + \lvert\mathcal{R}\rvert)d$
KBGAN [8]	GAN	reinforce learning (with pretrain)	$O(mN_1 d)$	$O(mN_1 d)$	$2(\lvert\mathcal{E}\rvert + \lvert\mathcal{R}\rvert)d$
NSCaching	using cache	gradient descent (from scratch)	$O(\frac{m}{n+1}(N_1 + N_2)d)$	$O(m(N_1 + N_2)d)$	$(\lvert\mathcal{E}\rvert + \lvert\mathcal{R}\rvert)d$

Comparison

GAN based	NSCaching
Increased number of training parameters	No extra parameters introduced
Sampling is not efficient	Efficient sampling through the cache
Training suffers from instability and degeneracy	Stable without pre-train

Outline

- Introduction of KG Embedding
- Problems in Negative Sampling
- NSCaching Frameworks
- Experiments
- Summary

Effectiveness

Measurements

- Given a triplet (h, r, t);
- Compute the score of $(h', r, t), \forall h' \in \mathcal{E}$;
- Get the rank of h among all h';
- Same for t.

Metrics

- MRR (mean reciprocal rank): $\frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\frac{1}{\text{rank}_i}$
- MR (mean rank): $\frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\text{rank}_i$
- Hit@10: $\frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\mathbb{I}(\text{rank}_i < 10)$

Datasets

	(#entities, #relations)
• WN18	(40943, 18)
• WN18RR	(40943, 11)
• FB15K	(14951, 1345)
• FB15K237	(14541, 237)

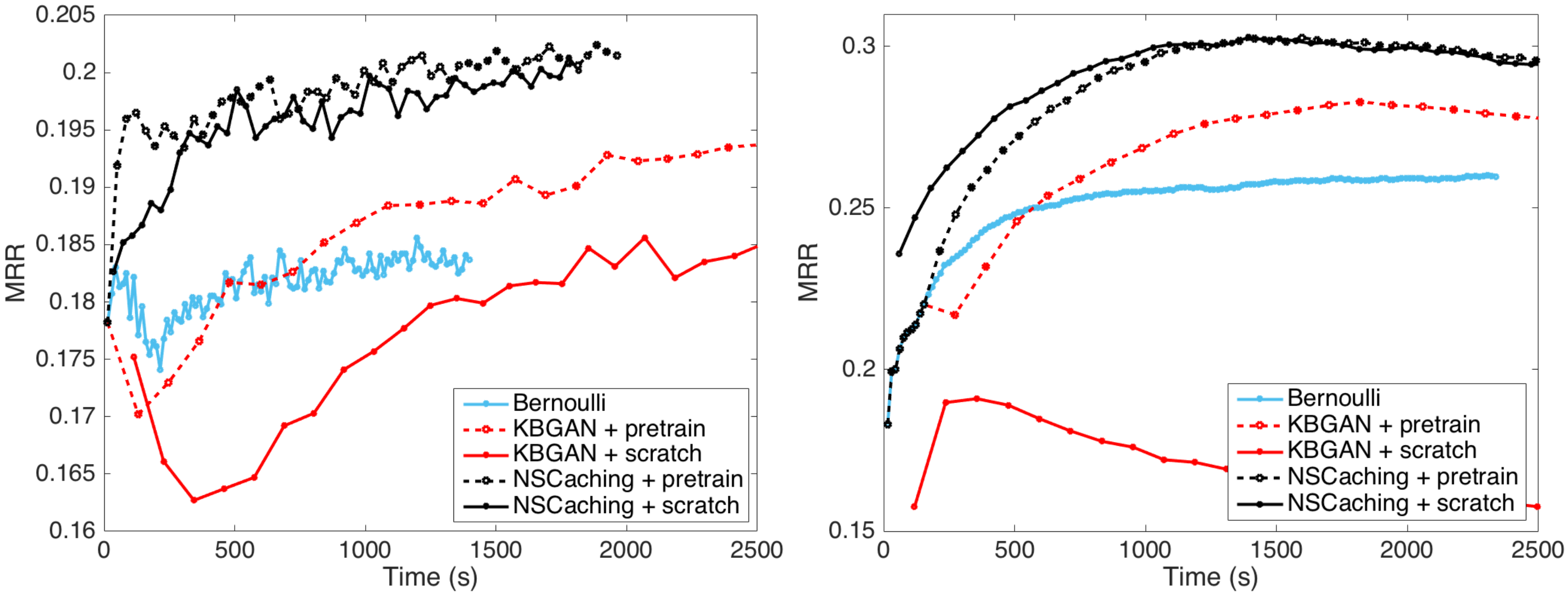
model	scoring function	definition
translational distance	TransE [7]	$\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_1$
	TransH [36]	$\|\mathbf{h}-\mathbf{w}_r^\top\mathbf{h}\mathbf{w}_r+\mathbf{r}-(\mathbf{t}-\mathbf{w}_r^\top\mathbf{t}\mathbf{w}_r)\|_1$
	TransD [16]	$\|\mathbf{h}+\mathbf{w}_r\mathbf{w}_h^\top\mathbf{h}+\mathbf{r}-(\mathbf{t}+\mathbf{w}_r\mathbf{w}_t^\top\mathbf{t})\|_1$
semantic matching	DistMult [38]	$\langle\mathbf{h},\mathbf{r},\mathbf{t}\rangle$
	ComplEx [32]	$\text{Re}\left(\langle\mathbf{h},\mathbf{r},\text{conj}(\mathbf{t})\rangle\right)$

Performance on *ComplEx*. **Bold** is best, underline is second best.

	WN18RR			FB15K237		
	MRR	MR	Hit10	MRR	MR	Hit10
Bernoulli	0.4431	**4693**	**51.77**	0.2596	238	43.54
KBGAN pretrain	0.4287	6929	47.03	0.2818	268	45.54
KBGAN scratch	0.3180	7528	33.51	0.1910	881	32.07
NSCaching pretrain	**0.4487**	<u>4861</u>	<u>51.76</u>	<u>0.3017</u>	**220**	<u>47.75</u>
NSCaching scratch	<u>0.4463</u>	5365	50.89	**0.3021**	<u>221</u>	**48.05**

Efficiency

We measure the convergence by *testing performance v.s. training time*.



TransD on WN18RR

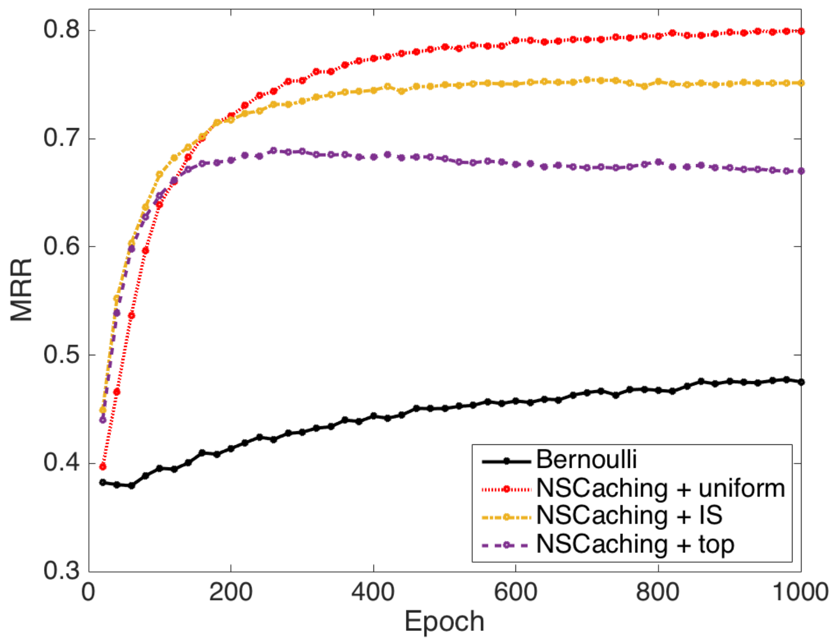
ComplEx on FB15K237

Updating and sampling time is included.

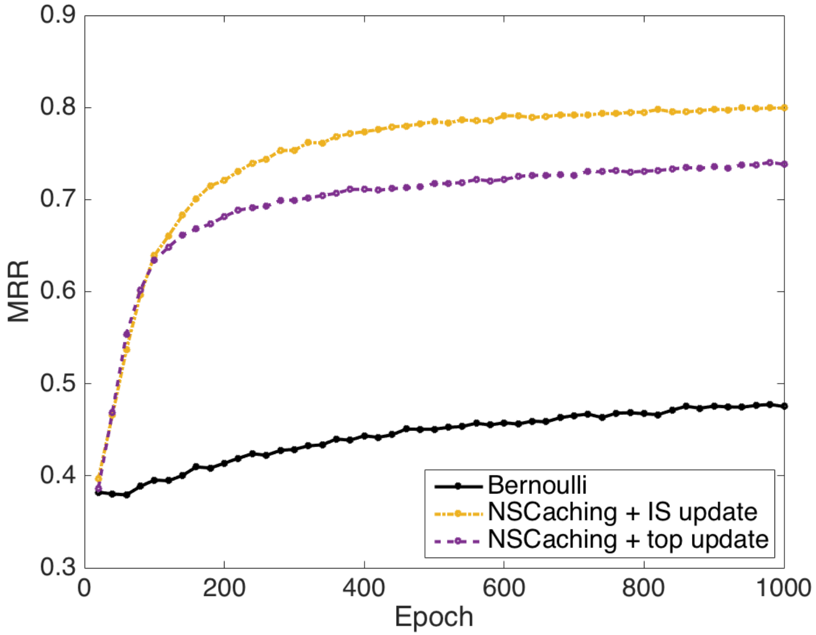
Sampling and updating schemes

Sampling from cache: uniform, importance sampling (IS), top-1

Cache update: importance sampling (IS), top-k



Diff. sampling scheme.

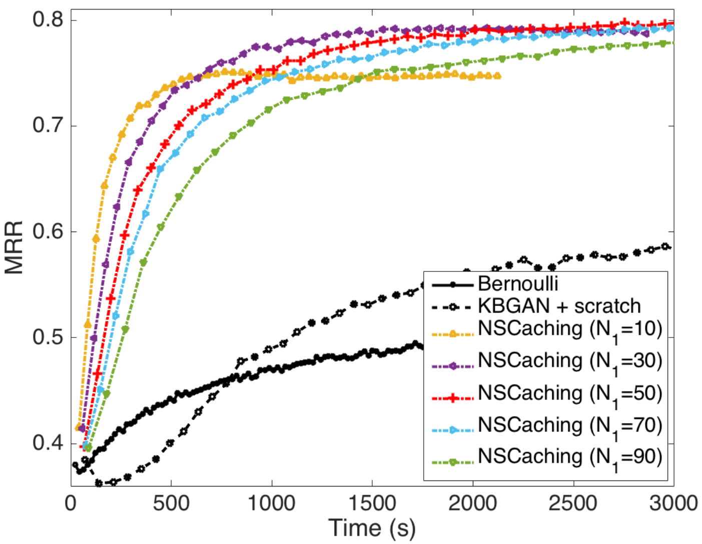


Diff. updating scheme.

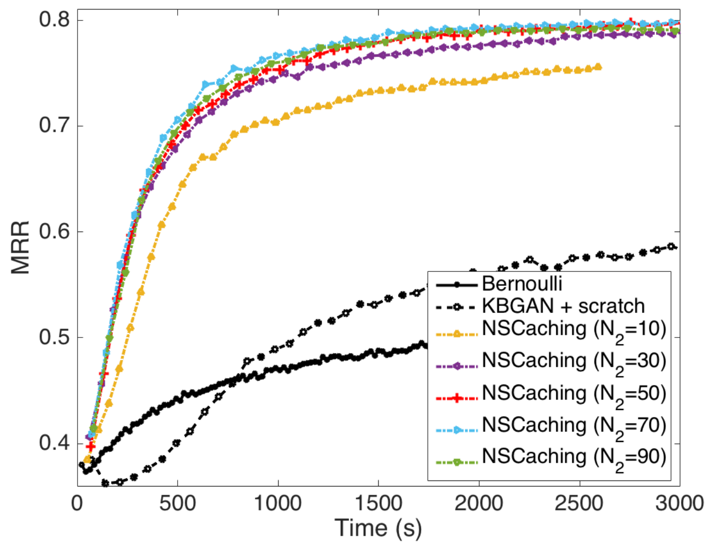
Stability

We change the cache size N_1 among $\{10, 30, 50, 70, 90\}$ when fixing $N_2 = 50$,

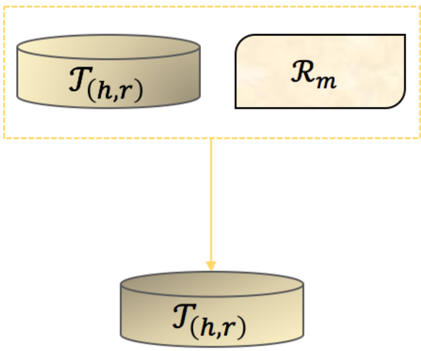
and random subset size N_2 among $\{10, 30, 50, 70, 90\}$ when fixing $N_1 = 50$.



Diff. N_1



Diff. N_2



Visualization

Given positive triplet *(manorama, profession, actor)*, we randomly select and visualize some entities in the tail-cache $\mathcal{T}_{(manorama,profession)}$ during training.

epoch	entities in cache
0	*allen_clarke, jose_gola, ostrava, ben_lilly, hans_zinsser*
20	*accountant, frank_pais, laura_marx, como, domitia_lepida*
100	*artist, , aviator, hans_zinsse, john_h_cough*
200	*physician, artist, raich_carter, coach, mark_shivas*
500	*artist, physician, cavan, sex_worker, attorney_at_law*

easy

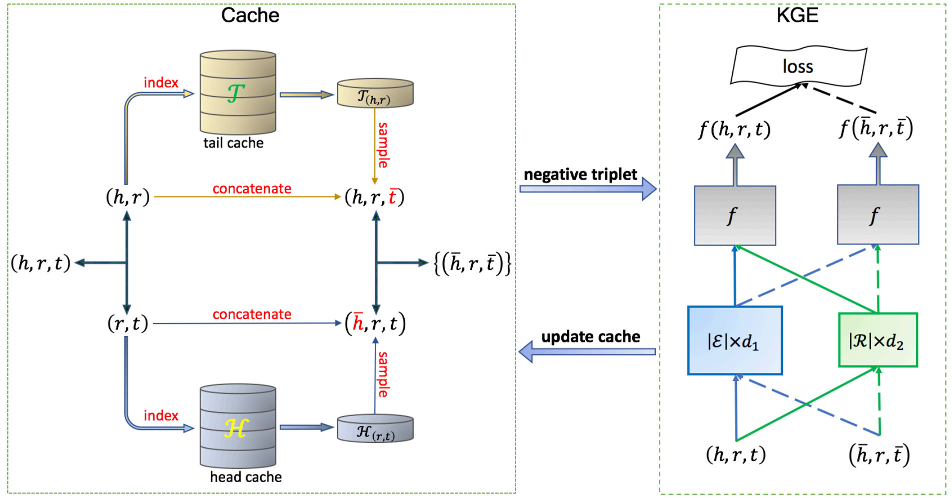
hard

Self-paced learning: from easy to gradually more complex examples.

Outline

- Introduction of KG Embedding
- Problems in Negative Sampling
- NSCaching Frameworks
- Experiments
- Summary

Summary



A novel negative sampling method.

Why it works

- It can dynamically hold high-quality negative samples;
- Sampling is efficient and extra memory is small;
- Both sampling and updating schemes are carefully designed to balance through exploration and exploitation;
- The cache schemes has connection with self-paced learning.

Future work

- Word/Network embedding.
- Advanced data structure to improve efficiency on extremely large scale KG.

Thank You

Code: https://github.com/yzhangee/NSCaching