Automated Knowledge Graph Embedding

Yongqi ZHANG

Researcher@4Paradigm

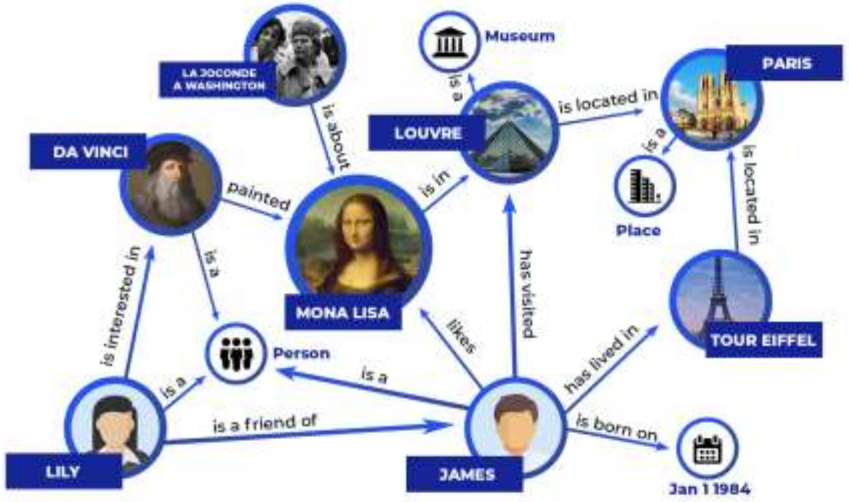
Joint work with Quanming YAO

August 23th, 2020

Outline

- **Introduction and Background**
- **AutoSF: Automated Scoring Function**
- **NRASE: NAS for Relational Path**
- **Summary**

Knowledge graph



- **Graph representation:** $\mathcal{G} = (E, R, S)$.
- **Entities** E: real world objects or abstract concepts.
- **Relations** R: interactions between/among entities.
- **Fact/triples** S: the basic unit in form of (head entity, relation, tail entity), (h, r, t).

- **Other related information:**
  - Types/attributes of entities/relations.
  - Text descriptions on entities and relations.
  - Ontologies: concept level description.
  - Logic rules: regular expressions.

Important properties

Semantic information

Symmetric, inverse, asymmetric, composition…

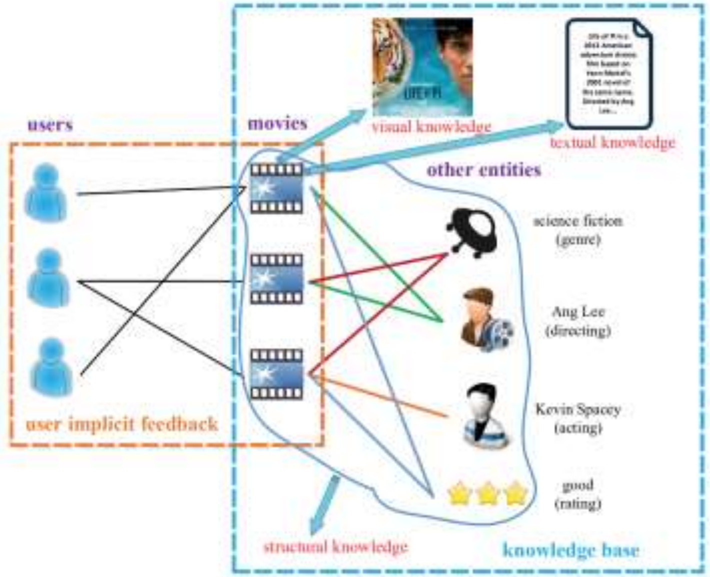
- (A, *spouse*, B) ⇔ (B, *spouse*, A)
- (A, *older*, B) ⇔ (B, *younger*, A)
- (A, *location*, USA)
- (A, *isBrotherOf*, B) ∧ (B, *isFatherOf*, C) ⇒ (A, *isUncleOf*, C)

Attribute information

- Indicate location, time, label, area, id, salary, …

Graph property

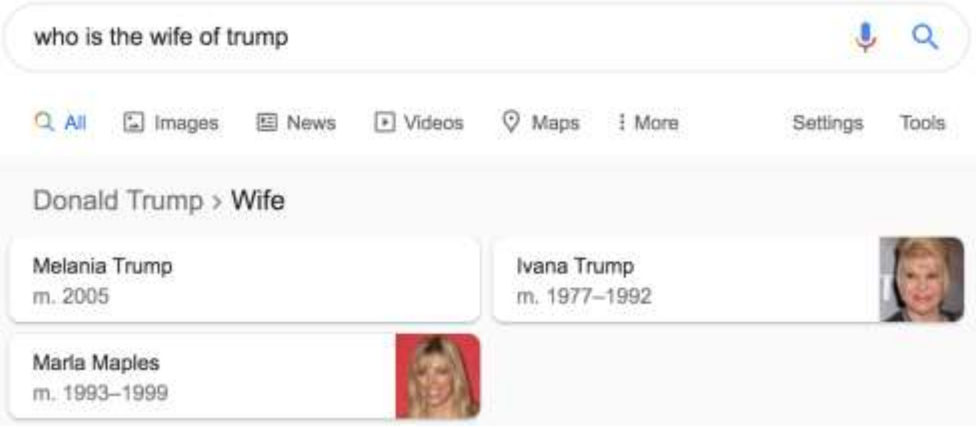
- A kind of heterogeneous information network.



Important applications

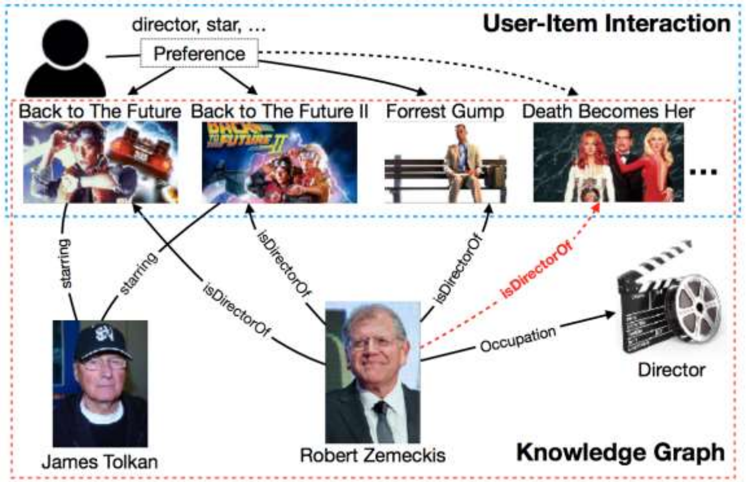
KGQA:

natural language -> query language
-> concise answer in KG.



Recommendation:

Improve accuracy and interpretability.



Medical diagnostic:

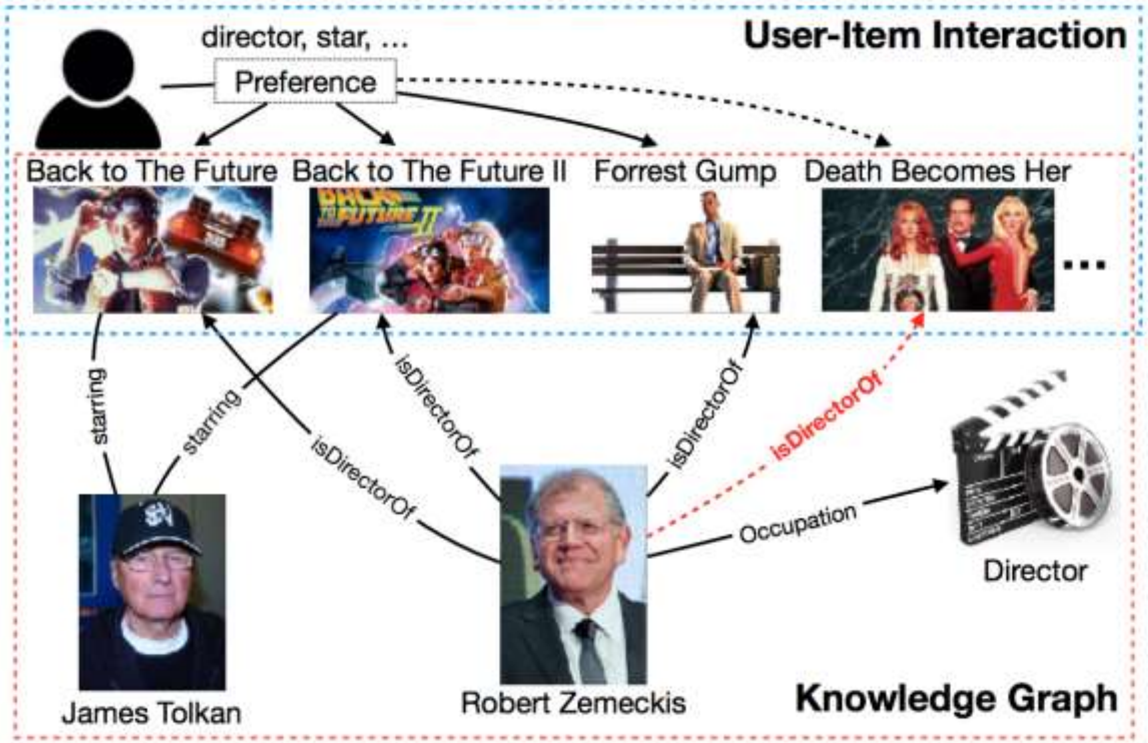
Get disease related suggestions.

Anti-fraud:

When fraud happens, who is the most related .

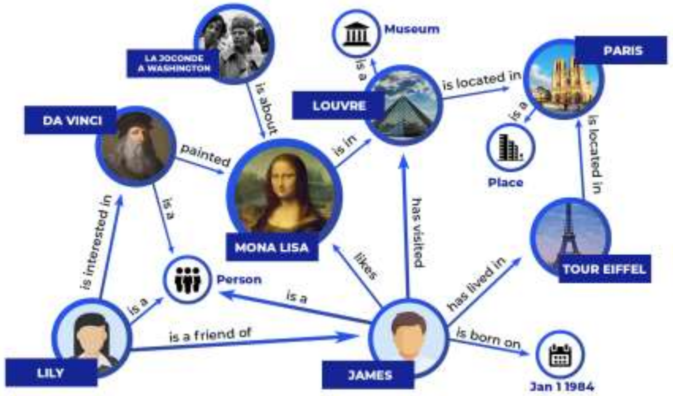
KG for recommendation

- User-user connections:
  - Social relationships
  - `(Tom,isFriendof,Bob)`
- User-item interactions:
  - `(user,clicks,item)`
  - `(user,prefers,item)`
- Item-item interactions:
  - Attributes, additional information
  - `(Robert,isDirectorOf,ForrestGump)`
- Benefits:
  - Rich semantic, structural information on items.
  - Explore user interests reasonably and offer explanations.

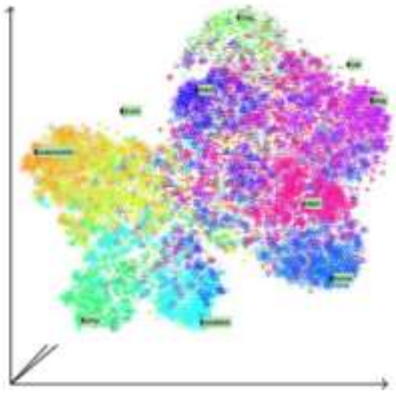


Learning KG embeddings

Encode entities and relations in KG into low-dimensional vector spaces $\mathbb{R}^{d_1}$ and $\mathbb{R}^{d_2}$, while capturing nodes' and edges' connection properties.



input



output

➢Objectives:

$$\min_{\boldsymbol{w}} \left|\overbrace{\gamma - f(\underbrace{\boldsymbol{w}; S^{+}) + f(\boldsymbol{w};}_{\text{parameters}}}^{\text{model}} S^{-})\right| \xrightarrow[\text{optimization}]{\text{iterative}} \text{Improve performance}$$

Observed triplet S^{+}:
increase score

Unobserved triplet S^{-}:
decrease score

$f(\boldsymbol{v}_{user}, \boldsymbol{v}_{prefers}, \boldsymbol{v}_{item})$

Outline

- **Introduction and Background**
- **AutoSF: Automated Scoring Function**
  - Preliminaries
  - Proposed method
  - Empirical results
  - Short summary
- **NRASE: NAS for Relational Path**
- **Summary**

Scoring functions

- A large amount of scoring functions (SFs) $f(\mathbf{h},\mathbf{r},\mathbf{t})$ are defined to measure the plausibility of triplets $\{(h,r,t)\}$ in KG.
- A branch focuses on single triplet, another on relational path.

Summary of Translational Distance Models (See Section 3.1 for Details)

Method	Ent. embedding	Rel. embedding	Scoring function $f_r(h,t)$	Constraints/Regularization
TransE [14]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$			
TransH [15]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$			
TransR [16]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$			
TransD [50]	$\mathbf{h},\mathbf{w}_h\in\mathbb{R}^d$ $\mathbf{t},\mathbf{w}_t\in\mathbb{R}^d$			
TranSparse [51]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$			
TransM [52]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$			
ManifoldE [53]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$			
TransF [54]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$			
TransA [55]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$			
KG2E [45]	$\mathbf{h}\sim\mathcal{N}(\boldsymbol{\mu}_h,\boldsymbol{\Sigma}_h)$ $\mathbf{t}\sim\mathcal{N}(\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)$ $\boldsymbol{\mu}_h,\boldsymbol{\mu}_t\in\mathbb{R}^d$ $\boldsymbol{\Sigma}_h,\boldsymbol{\Sigma}_t\in\mathbb{R}^{d\times d}$			
TransG [46]	$\mathbf{h}\sim\mathcal{N}(\boldsymbol{\mu}_h,\sigma_h^2\mathbf{I})$ $\mathbf{t}\sim\mathcal{N}(\boldsymbol{\mu}_t,\sigma_t^2\mathbf{I})$ $\boldsymbol{\mu}_h,\boldsymbol{\mu}_t\in\mathbb{R}^d$			
UM [56]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$			
SE [57]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$			

Summary of Semantic Matching Models (See Section 3.2 for Details)

Method	Ent. embedding	Rel. embedding	Scoring function $f_r(h,t)$	Constraints/Regularization
RESCAL [13]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{M}_r\in\mathbb{R}^{d\times d}$	$\mathbf{h}^\top\mathbf{M}_r\mathbf{t}$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{M}_r\|_F\le 1$ $\mathbf{M}_r=\sum_i \pi_r^i\mathbf{u}_i\mathbf{v}_i^\top$ (required in [17])
TATEC [64]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d, \mathbf{M}_r\in\mathbb{R}^{d\times d}$	$\mathbf{h}^\top\mathbf{M}_r\mathbf{t}+\mathbf{h}^\top\mathbf{r}+\mathbf{t}^\top\mathbf{r}+\mathbf{h}^\top\mathbf{D}\mathbf{t}$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$ $\|\mathbf{M}_r\|_F\le 1$
DistMult [65]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$\mathbf{h}^\top\mathrm{diag}(\mathbf{r})\mathbf{t}$	$\|\mathbf{h}\|_2=1, \|\mathbf{t}\|_2=1, \|\mathbf{r}\|_2\le 1$
HolE [62]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$\mathbf{r}^\top(\mathbf{h}\star\mathbf{t})$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$
ComplEx [66]	$\mathbf{h},\mathbf{t}\in\mathbb{C}^d$	$\mathbf{r}\in\mathbb{C}^d$	$\mathrm{Re}(\mathbf{h}^\top\mathrm{diag}(\mathbf{r})\bar{\mathbf{t}})$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$
ANALOGY [68]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{M}_r\in\mathbb{R}^{d\times d}$	$\mathbf{h}^\top\mathbf{M}_r\mathbf{t}$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{M}_r\|_F\le 1$ $\mathbf{M}_r\mathbf{M}_r^\top=\mathbf{M}_r^\top\mathbf{M}_r$ $\mathbf{M}_r\mathbf{M}_{r'}=\mathbf{M}_{r'}\mathbf{M}_r$
SME [18]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$(\mathbf{M}_u^1\mathbf{h}+\mathbf{M}_u^2\mathbf{r}+\mathbf{b}_u)^\top(\mathbf{M}_v^1\mathbf{t}+\mathbf{M}_v^2\mathbf{r}+\mathbf{b}_v)$ $((\mathbf{M}_u^1\mathbf{h})\circ(\mathbf{M}_u^2\mathbf{r})+\mathbf{b}_u)^\top((\mathbf{M}_v^1\mathbf{t})\circ(\mathbf{M}_v^2\mathbf{r})+\mathbf{b}_v)$	$\|\mathbf{h}\|_2=1, \|\mathbf{t}\|_2=1$
NTN [19]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r},\mathbf{b}_r\in\mathbb{R}^k, \underline{\mathbf{M}}_r\in\mathbb{R}^{d\times d\times k}$ $\mathbf{M}_r^1,\mathbf{M}_r^2\in\mathbb{R}^{k\times d}$	$\mathbf{r}^\top\tanh(\mathbf{h}^\top\underline{\mathbf{M}}_r\mathbf{t}+\mathbf{M}_r^1\mathbf{h}+\mathbf{M}_r^2\mathbf{t}+\mathbf{b}_r)$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$ $\|\mathbf{b}_r\|_2\le 1, \|\underline{\mathbf{M}}_r^{[:,:,i]}\|_F\le 1$ $\|\mathbf{M}_r^1\|_F\le 1, \|\mathbf{M}_r^2\|_F\le 1$
SLM [19]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^k, \mathbf{M}_r^1,\mathbf{M}_r^2\in\mathbb{R}^{k\times d}$	$\mathbf{r}^\top\tanh(\mathbf{M}_r^1\mathbf{h}+\mathbf{M}_r^2\mathbf{t})$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$ $\|\mathbf{M}_r^1\|_F\le 1, \|\mathbf{M}_r^2\|_F\le 1$
MLP [69]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$\mathbf{w}^\top\tanh(\mathbf{M}^1\mathbf{h}+\mathbf{M}^2\mathbf{r}+\mathbf{M}^3\mathbf{t})$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$
NAM [63]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$f_r(h,t)=\mathbf{t}^\top\mathbf{z}^{(L)}$ $\mathbf{z}^{(\ell)}=\mathrm{ReLU}(\mathbf{a}^{(\ell)}),\ \mathbf{a}^{(\ell)}=\mathbf{M}^{(\ell)}\mathbf{z}^{(\ell-1)}+\mathbf{b}^{(\ell)}$ $\mathbf{z}^{(0)}=[\mathbf{h};\mathbf{r}]$	—

[Wang et. al. TKDE 2017]

Important properties

Given (h, r, t), the reversed triplet is (t, r, h).

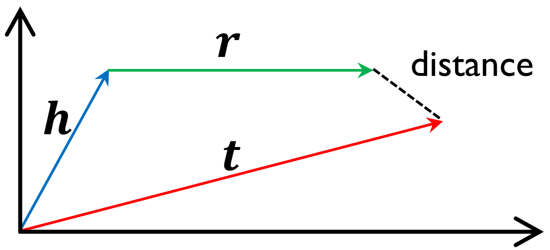
Common relations	Requirements on f	Examples
symmetric	$f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) = f(\boldsymbol{t}, \boldsymbol{r}, \boldsymbol{h})$	*IsSimilarTo*, *Spouse*
anti-symmetric	$f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) = -f(\boldsymbol{t}, \boldsymbol{r}, \boldsymbol{h})$	*LargerThan*, *Hypernym*
general asymmetric	$f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) \neq f(\boldsymbol{t}, \boldsymbol{r}, \boldsymbol{h})$	*LocatedIn*, *Profession*
inverse	$f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) = f(\boldsymbol{t}, \boldsymbol{r}', \boldsymbol{h})$	(*Hypernym*, *Hyponym*)
composition	-	$Father \circ Spouse \rightarrow Mother$

Outline

- ➢ **Introduction and Background**
- ➢ **AutoSF: Automated Scoring Function**
  - Preliminaries
  - Proposed method
  - Empirical results
  - Short summary
- ➢ **NRASE: NAS for Relational Path**
- ➢ **Summary**

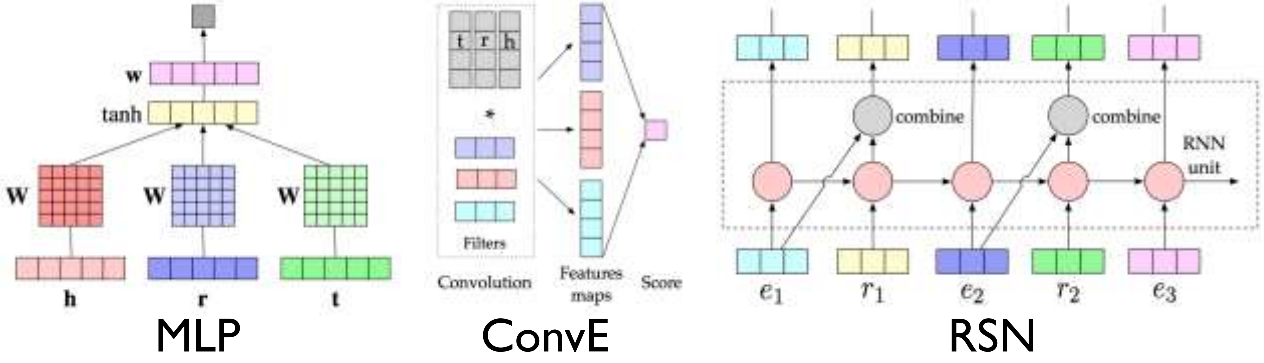
General types

- ➢ Translation Distance Models (TDMs)
  - TransE, TransH, RotatE, etc.
  - less expressive. [Wang et. al. AAAI 2017]



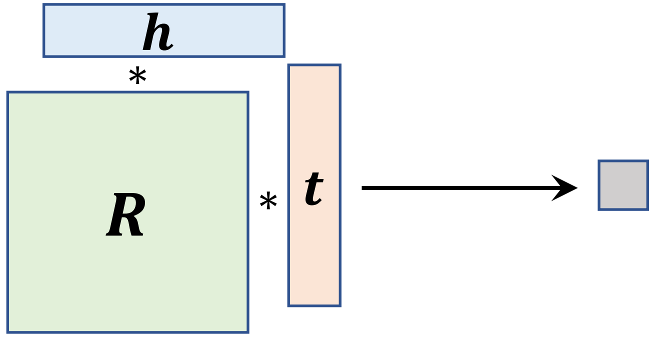
- ➢ Neural Network Models (NNMs)
  - MLP, ConvE, RSN, etc.
  - complex and difficult to train.

    [Wang et. al. TKDE 2017]



- ➢ BiLinear Models (BLMs)
  - DistMult, ComplEx, Analogy, SimplE, etc.
  - state-of-the-art and fully expressive.

    [Wang et. al. AAAI 2017], [Lacroix et. al. ICML 2018]



Graphical illustration of BLMs

The BLMs can be written as $f(\mathbf{h},\mathbf{r},\mathbf{t}) = \mathbf{h}^{\mathrm{T}}\mathbf{R}\mathbf{t}$, with different form of $\mathbf{R}$, a square matrix of $\mathbf{r}$.
For unified representation, we evenly split the embedding into 4 parts, e.g. $\boldsymbol{r} = [\boldsymbol{r}_1; \boldsymbol{r}_2; \boldsymbol{r}_3; \boldsymbol{r}_4]$.
Denote $\mathbf{D}_i^r = \mathrm{diag}(\boldsymbol{r}_i)$ as the corresponding diagonal matrix.

DistMult: $f(h,r,t) = \langle \mathbf{h}, \mathbf{r}, \mathbf{t} \rangle$

$$\begin{bmatrix} \mathbf{D}_1^r & & & \\ & \mathbf{D}_2^r & & \\ & & \mathbf{D}_3^r & \\ & & & \mathbf{D}_4^r \end{bmatrix}$$

ComplEx: $f(h,r,t) = \mathrm{Re}(\langle \mathbf{h}, \mathbf{r}, \mathrm{conj}(\mathbf{t}) \rangle)$

$$\begin{bmatrix} \mathbf{D}_1^r & & \mathbf{D}_3^r & \\ & \mathbf{D}_2^r & & \mathbf{D}_4^r \\ -\mathbf{D}_3^r & & \mathbf{D}_1^r & \\ & -\mathbf{D}_4^r & & \mathbf{D}_2^r \end{bmatrix}$$

Analogy: $f(h,r,t) = \langle \hat{\mathbf{h}}, \hat{\mathbf{r}}, \hat{\mathbf{t}} \rangle + \mathrm{Re}\big(\langle \breve{\mathbf{h}}, \breve{\mathbf{r}}, \mathrm{conj}(\breve{\mathbf{t}}) \rangle\big)$

$$\begin{bmatrix} \mathbf{D}_1^r & & & \\ & \mathbf{D}_2^r & & \\ & & \mathbf{D}_3^r & \mathbf{D}_4^r \\ & & -\mathbf{D}_4^r & \mathbf{D}_3^r \end{bmatrix}$$

SimplE: $f(h,r,t) = \langle \hat{\mathbf{h}}, \hat{\mathbf{r}}, \breve{\mathbf{t}} \rangle + \langle \breve{\mathbf{h}}, \breve{\mathbf{r}}, \hat{\mathbf{t}} \rangle$

$$\begin{bmatrix} & & \mathbf{D}_1^r & \\ & & & \mathbf{D}_2^r \\ \mathbf{D}_3^r & & & \\ & \mathbf{D}_4^r & & \end{bmatrix}$$

Key problems

1. There is no absolute winner among them since KGs exhibit distinct patterns. Even the fully expressive models do not definitely perform the best.
2. KG is sparse, thus regularization is important.
3. Designing novel and universal SFs becomes harder.

Our solutions:

- Adaptively search how to regularize the BLMs for different KG tasks.
- Design novel and task-aware scoring functions.

AutoSF

Definition 1 (AutoSF). *Let $F(g)$ be a KGE model (with indexed embeddings $\mathbf{h}, \mathbf{r}, \mathbf{t}$ and structure g). $\mathcal{M}(F(g), \mathcal{S})$ measures the performance (the higher the better) of a KGE model F with on a set of triplets $\mathcal{S}$. The problem of searching the SF is formulated as:*

$$\boxed{g^* \in \arg\max_{g \in \mathcal{G}} \mathcal{M}\left(F^*(g), \mathcal{S}_{val}\right)} \tag{1}$$

$$\text{s.t. } \boxed{F^*(g) = \arg\max_F \mathcal{M}(F(g), \mathcal{S}_{tra}),} \tag{2}$$

where $\mathcal{G}$ contains all possible choices of g, $\mathcal{S}_{tra}$ and $\mathcal{S}_{val}$ denote training and validation sets.

Search space:
What to be searched

$\boxed{g \in \mathcal{G}}$

$F^*(g) = \arg\max_{F(g)} \mathcal{M}(F(g), S_{tra})$

Validation performance
$\mathcal{M}(F^*(g), S_{val})$

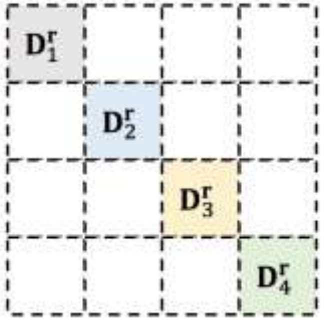
Structure optimizer $\boxed{\max_{g \in \mathcal{G}} \mathcal{M}(F^*(g), S_{val})}$

Search algorithm:
How to search efficiently

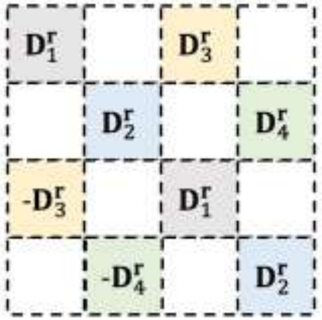
Search space

Definition 2 (Search space). *Let $g(\mathbf{r})$ return a 4×4 block matrix, of which the elements in each block is given by $[g(\mathbf{r})]_{ij} = diag(\mathbf{a}_{ij})$ where $\mathbf{a}_{ij} \in \{\mathbf{0}, \pm\mathbf{r}_1, \pm\mathbf{r}_2, \pm\mathbf{r}_3, \pm\mathbf{r}_4\}$ for $i, j \in \{1, 2, 3, 4\}$. Then, SFs can be represented by $f_{unified}(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \sum_{i,j} \langle \mathbf{h}_i, \mathbf{a}_{ij}, \mathbf{t}_j \rangle = \mathbf{h}^\top g(\mathbf{r})\, \mathbf{t}$.*

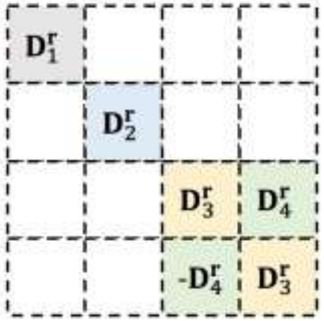
The location of a block matrix $\mathbf{D}_i^r$ represents a multiplicative term.



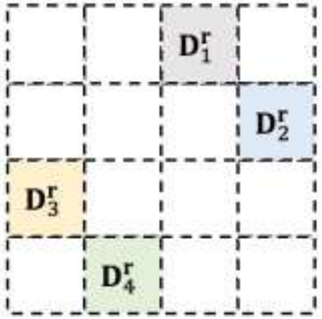
DistMult



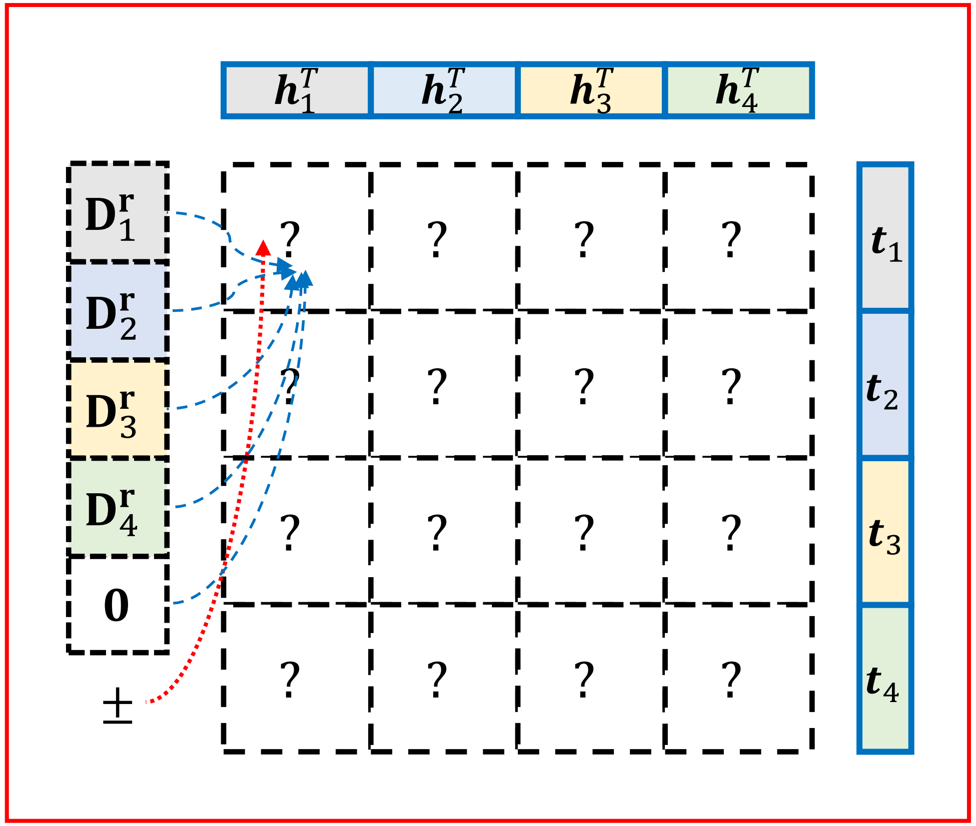
ComplEx



Analogy



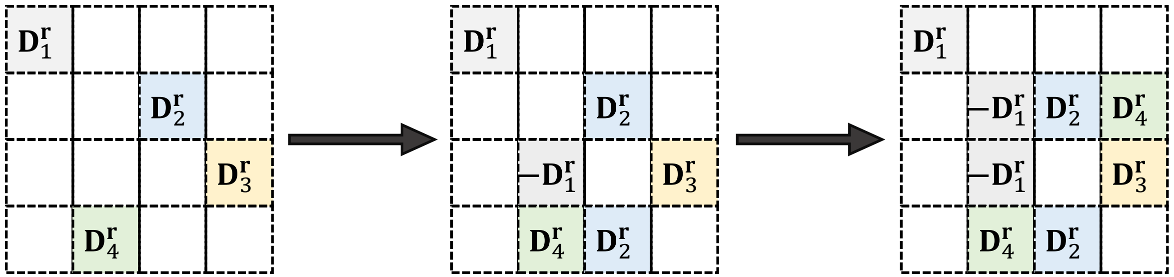
SimplE



9^{16} candidates!

Search algorithm

- ➢Greedy search: progressively evaluate from few blocks to more blocks.



For f^6, reduces from 2×10^9 to 3×10^4.

- ➢Filter: remove bad and equivalent SFs.

For f^4, reduces from 9216 to 5.

- ➢Predictor: select promising SFs based on matrix structures.
  - The predictor learns a mapping from structure to performance.

Select $K_2 = 8$ from $N = 256$.

Key idea: select better SFs based on matrix structure to train and evaluate.

Outline

- **Introduction and Background**
- **AutoSF: Automated Scoring Function**
  - Preliminaries
  - Proposed method
  - Empirical results
  - Short summary
- **NRASE: NAS for Relational Path**
- **Summary**

Effectiveness

type	model	WN18		FB15k		WN18RR		FB15k237		YAGO3-10	
		MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10
TDM	TransE [51]	0.500	94.1	0.495	77.4	0.178	45.1	0.256	41.9	—	—
	TransH [51]	0.521	94.5	0.452	76.6	0.186	45.1	0.233	40.1	—	—
	RotatE [34]	0.949	95.9	0.797	88.4	0.476	**57.1**	0.338	53.3	—	—
NNM	NTN [46]	0.53	66.1	0.25	41.4	—	—	—	—	—	—
	Neural LP [47]	0.94	94.5	0.76	83.7	—	—	0.24	36.2	—	—
	ConvE [6]	0.94	95.6	0.745	87.3	0.46	48	0.325	50.1	0.52	66.0
BLM	TuckER [1]	**0.953**	95.8	0.795	89.2	0.470	52.6	0.358	54.4	—	—
	HolEX [45]	0.938	94.9	0.800	88.6	—	—	—	—	—	—
	DistMult	0.821	95.2	0.817	89.5	0.443	50.7	0.349	53.7	0.552	69.4
	ComplEx	0.951	95.7	0.831	90.5	0.471	55.1	0.347	54.1	0.566	70.9
	Analogy	0.950	95.7	0.829	90.5	0.472	55.8	0.348	54.7	0.565	71.3
	SimplE/CP	0.950	95.9	0.830	90.3	0.468	55.2	0.350	54.4	0.565	71.0
AutoSF		0.952	**96.1**	**0.861**	**91.4**	**0.490**	56.7	**0.365**	**55.5**	**0.582**	**71.7**

- BLMs are better than the other types.
- There is no absolute winner among the BLMs.
- Compared with human-designed ones, the SFs searched by AutoSF always lead the performance.

Distinctiveness

$$\begin{bmatrix} & \mathbf{D}^r_2 & -\mathbf{D}^r_2 & \mathbf{D}^r_1 \\ \mathbf{D}^r_3 & & \mathbf{D}^r_2 & \\ -\mathbf{D}^r_3 & \mathbf{D}^r_3 & & -\mathbf{D}^r_1 \\ \mathbf{D}^r_4 & & -\mathbf{D}^r_4 & \end{bmatrix}$$

(a) WN18.

$$\begin{bmatrix} & \mathbf{D}^r_2 & & \mathbf{D}^r_1 \\ \mathbf{D}^r_3 & & \mathbf{D}^r_2 & \\ & \mathbf{D}^r_3 & & \\ \mathbf{D}^r_4 & & & \end{bmatrix}$$

(b) FB15k.

$$\begin{bmatrix} \mathbf{D}^r_4 & \mathbf{D}^r_1 & & \\ \mathbf{D}^r_2 & \mathbf{D}^r_4 & & -\mathbf{D}^r_1 \\ & & \mathbf{D}^r_3 & -\mathbf{D}^r_1 \\ & -\mathbf{D}^r_2 & -\mathbf{D}^r_2 & \mathbf{D}^r_4 \end{bmatrix}$$

(c) WN18RR.

$$\begin{bmatrix} & & \mathbf{D}^r_1 & \\ & \mathbf{D}^r_2 & & \\ & & & \mathbf{D}^r_3 \\ \mathbf{D}^r_4 & & & \end{bmatrix}$$

(d) FB15k237.

$$\begin{bmatrix} \mathbf{D}^r_1 & & & \\ & \mathbf{D}^r_2 & & -\mathbf{D}^r_3 \\ & & \mathbf{D}^r_3 & \\ & -\mathbf{D}^r_2 & & \mathbf{D}^r_4 \end{bmatrix}$$

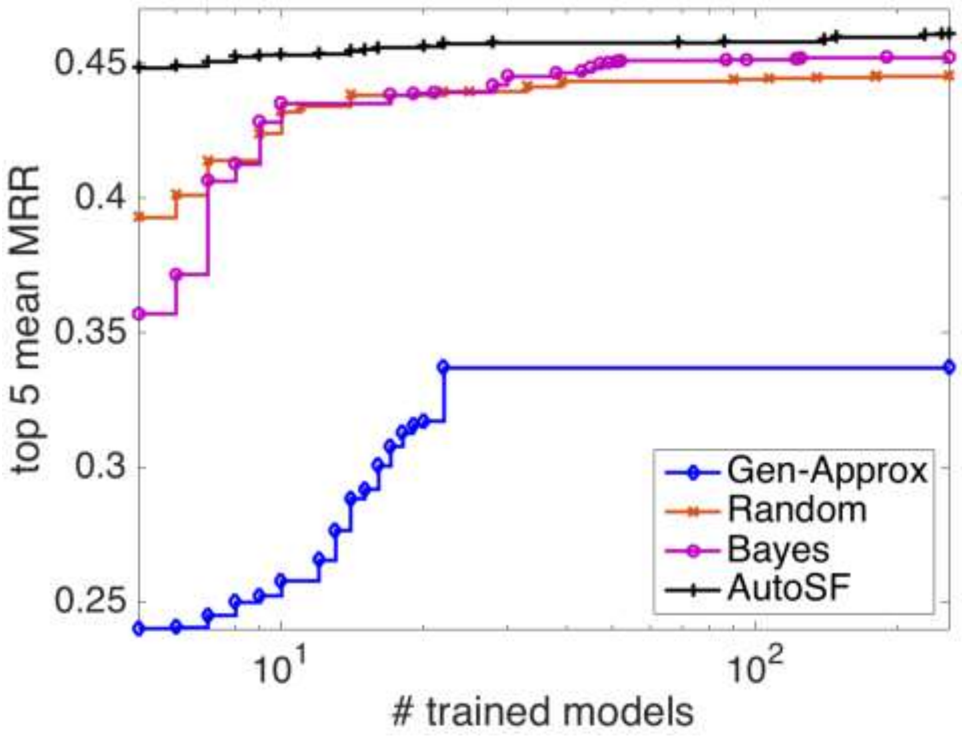
(e) YAGO3-10.

Table 4: MRRs of applying SF searched from one dataset (indicated by each row) on another dataset (indicated by each column).

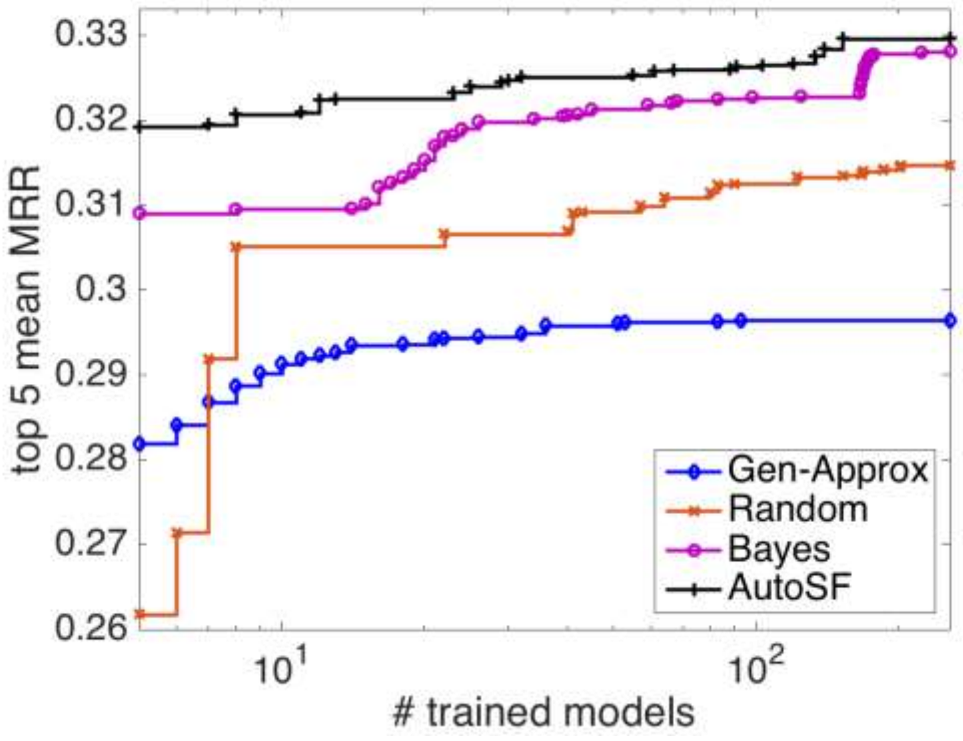
	WN18	FB15k	W-RR	F-237	YA-10
WN18	**0.952**	0.852	0.483	0.349	0.572
FB15k	0.950	**0.861**	0.481	0.350	0.574
WN18RR	0.951	0.849	**0.490**	0.345	0.574
FB15k237	0.894	0.781	0.471	**0.365**	0.571
YAGO3-10	0.885	0.844	0.476	0.352	**0.582**

The searched SFs are KG dependent and novel to the literature.

Efficiency



WN18RR



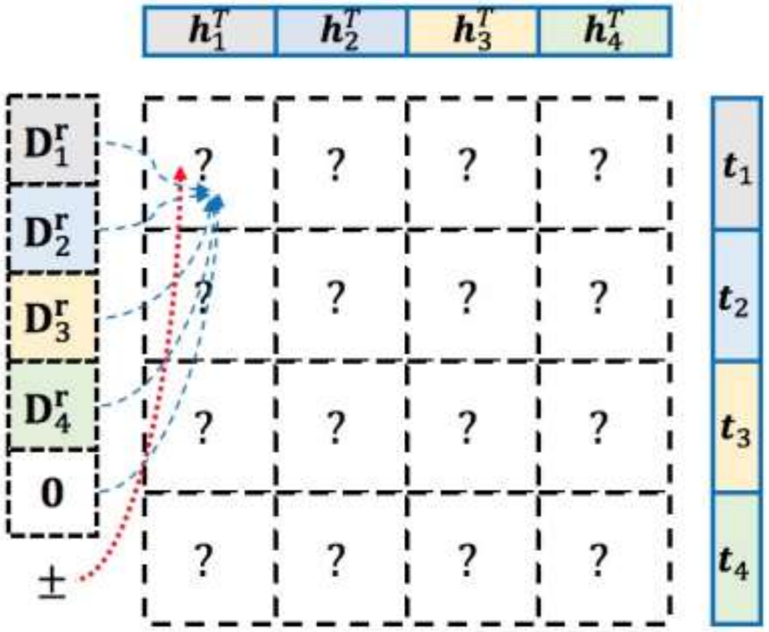
FB15k237

- Gen-Approx: a universal approximator MLP as the search space.
- Random: totally random for SF generation.
- Bayes: Tree Parzen Estimator (TPE) algorithm.
- AutoSF: domain-specific search algorithm.

Outline

- Introduction and Background
- AutoSF: Automated Scoring Function
  - Preliminaries
  - Proposed method
  - Empirical results
  - Short summary
- NRASE: NAS for Relational Path
- Summary

Summary of AutoSF



Challenges:

- Designing new and universal SFs are non-trivial.
- Different KG has distinct properties.

Contributions:

- The first AutoML work in SF design.
- Well-defined search space and search algorithm with domain knowledge.
- Task-aware SFs are searched efficiently.

- AutoSF: Searching Scoring Functions for Knowledge Graph Embedding. **Y. Zhang**, Q. Yao, W. Dai, L. Chen. ***ICDE 2020***

Outline

- **Introduction and Background**
- **AutoSF: Automated Scoring Function**
- **NRASE: NAS for Relational Path**
  - Preliminaries
  - Proposed method
  - Empirical results
  - Short summary
- **Summary**

Relational path & Path distiller

DEFINITION 1 (RELATIONAL PATH [17, 20]). *A path (of length L) is formed by a set of triplets $(s_1, r_1, o_1), (s_2, r_2, o_2), \ldots, (s_L, r_L, o_L)$ where $o_i = s_{i+1}$ for all $i = 1 \ldots L-1$.*

Semantic information:
- Preserved in the triplets.

Structural information:
- Preserved along the path.

$$s_1 \xrightarrow{r_1} s_2 \xrightarrow{r_2} \cdots \boxed{s_t \xrightarrow{r_t}} \cdots \xrightarrow{r_L} o_L$$

DEFINITION 2 (PATH DISTILLER). *A path distiller processes the embeddings of $\mathbf{s}_1, \mathbf{r}_1$ to $\mathbf{s}_L, \mathbf{r}_L$ recurrently. In each recurrent step t, the distiller combines embeddings of $\mathbf{s}_t$, $\mathbf{r}_t$ and a distillation of preceding information $\mathbf{h}_{t-1}$ to get an output $\mathbf{v}_t$. The distiller is formulated as a recurrent function*

$$[\mathbf{v}_t, \mathbf{h}_t] = f(\mathbf{s}_t, \mathbf{r}_t, \mathbf{h}_{t-1}), \quad t = 1 \ldots L, \tag{1}$$

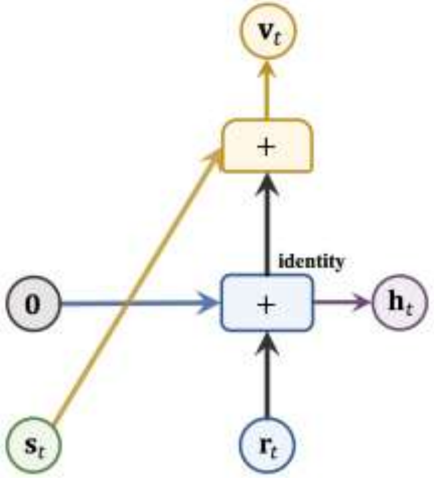
where $\mathbf{h}_t$'s are hidden state of recurrent steps and $\mathbf{h}_0 = \mathbf{s}_1$. The output $\mathbf{v}_t$ should approach object entity $\mathbf{o}_t$.

Meanings	Notations
head/subject entity	s_t
relation	r_t
tail/object entity	o_t
hidden state	$\mathbf{h}_t$

Outline

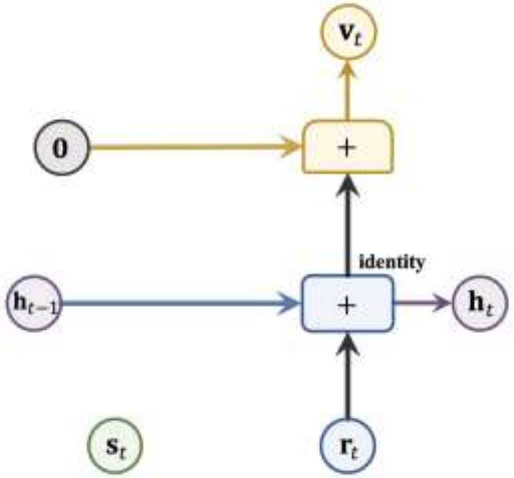
- **Introduction and Background**
- **AutoSF: Automated Scoring Function**
- **NRASE: NAS for Relational Path**
  - Preliminaries
  - Proposed method
  - Empirical results
  - Short summary
- **Summary**

Existing models



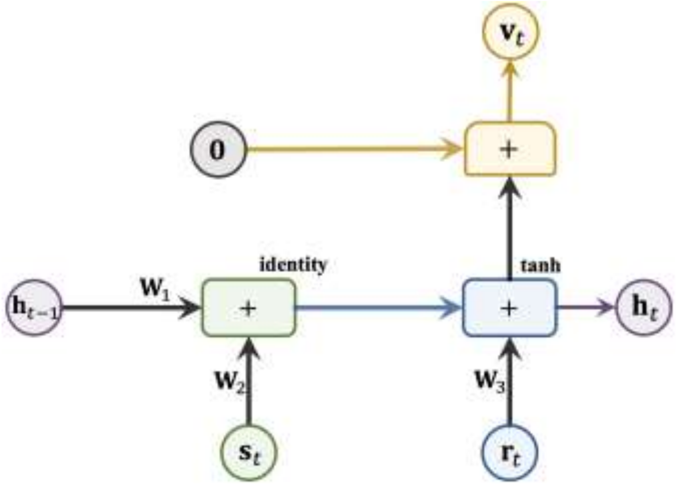
TransE

[Bordes et al. NIPS 2013]



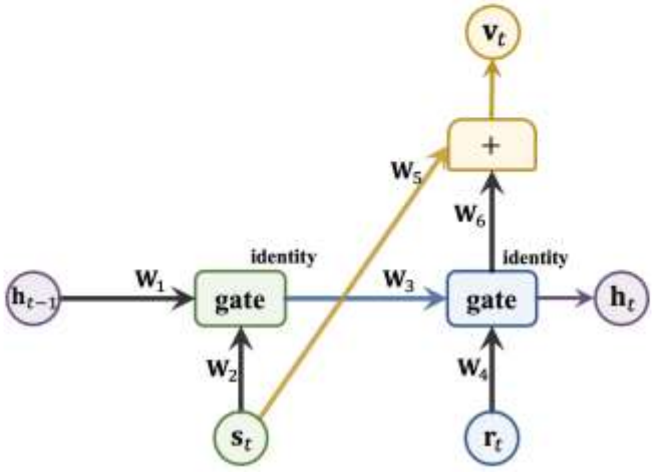
PTransE

[Lin et al. ACL 2015]



ChainR

[Das et al. ACL 2017]



RSN

[Guo et al. ICML 2019]

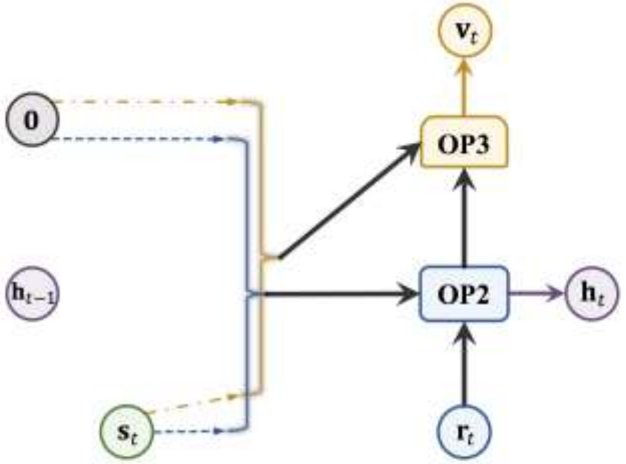
model		path-based	semantic	structural
TransE [7]		×	$\mathbf{v}_t = \mathbf{s}_t + \mathbf{r}_t$	×
ComplEx [45]		×	$\mathbf{v}_t = \mathbf{s}_t \otimes \mathbf{r}_t$	×
PTransE [25], TransE-Comp[19]	add	√	$\mathbf{v}_t = \mathbf{h}_t$	$\mathbf{h}_t = \mathbf{h}_{t-1} + \mathbf{r}_t$
	multiply	√	$\mathbf{v}_t = \mathbf{h}_t$	$\mathbf{h}_t = \mathbf{h}_{t-1} \odot \mathbf{r}_t$
	RNN	√	$\mathbf{v}_t = \mathbf{h}_t$	$\mathbf{h}_t = \sigma(\mathbf{W}_1\mathbf{r}_t + \mathbf{W}_2\mathbf{h}_{t-1} + \mathbf{b})$
ChainR [11]		√	$\mathbf{v}_t = \mathbf{h}_t$	$\mathbf{h}_t = \sigma(\mathbf{W}_1\mathbf{h}_{t-1} + \mathbf{W}_2\mathbf{s}_t + \mathbf{W}_3\mathbf{r}_t + \mathbf{b})$
RSN [17]		√	$\mathbf{v}_t = \mathbf{W}_5\mathbf{s}_t + \mathbf{W}_6\mathbf{h}_t$	$\mathbf{h}_t = \sigma(\mathbf{W}_3\sigma(\mathbf{W}_1\mathbf{h}_{t-1} + \mathbf{W}_2\mathbf{s}_t + \mathbf{b}_1) + \mathbf{W}_4\mathbf{r}_t + \mathbf{b}_2)$
NRASE		√	a recurrent network searched by natural gradient descent	

Key challenges

1. Different model performs differently on tasks.
   - Single triplet based models are expressive in link prediction tasks.
   - Composition patterns can be learned only when the path have strong semantic meaning.
   - Long-term information in entity alignment tasks.
2. Structural and semantic information are complex among different KGs.

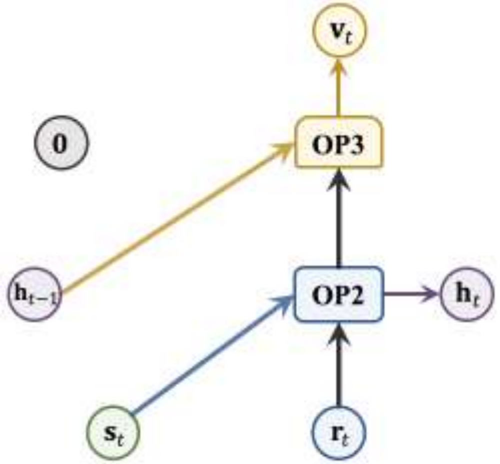
- ➢ How to distill the structural information from relational path and combine it with semantic information?
- ➢ Our solution:
  - Design a specific recurrent search space to cover existing methods;
  - Adaptively search the model for specific tasks.

Case study



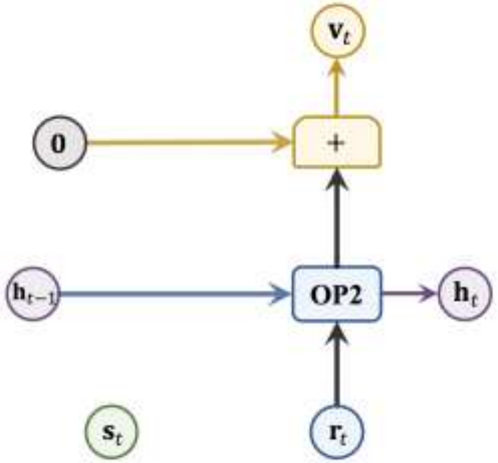
P1

single triplet



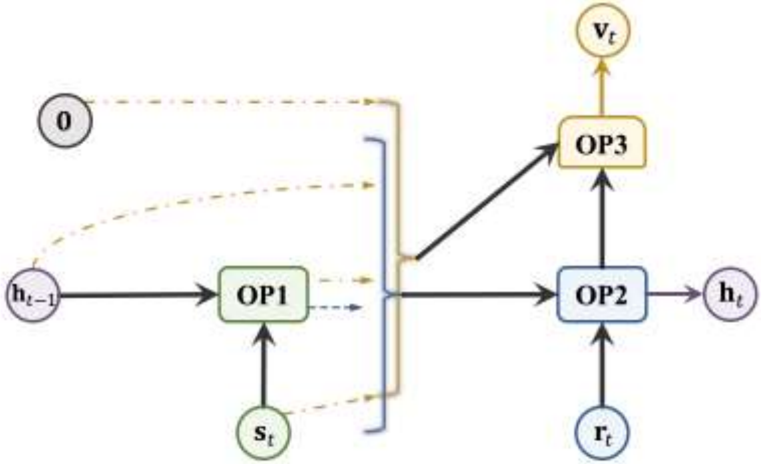
P2

two triplets



P3

composition pattern



P4

fully recurrent

data	tasks
S1	neighbor ∧ locatedin → locatedin
S2	neighbor ∧ locatedin → locatedin
S3	neighbor ∧ locatedin ∧ locatedin → locatedin

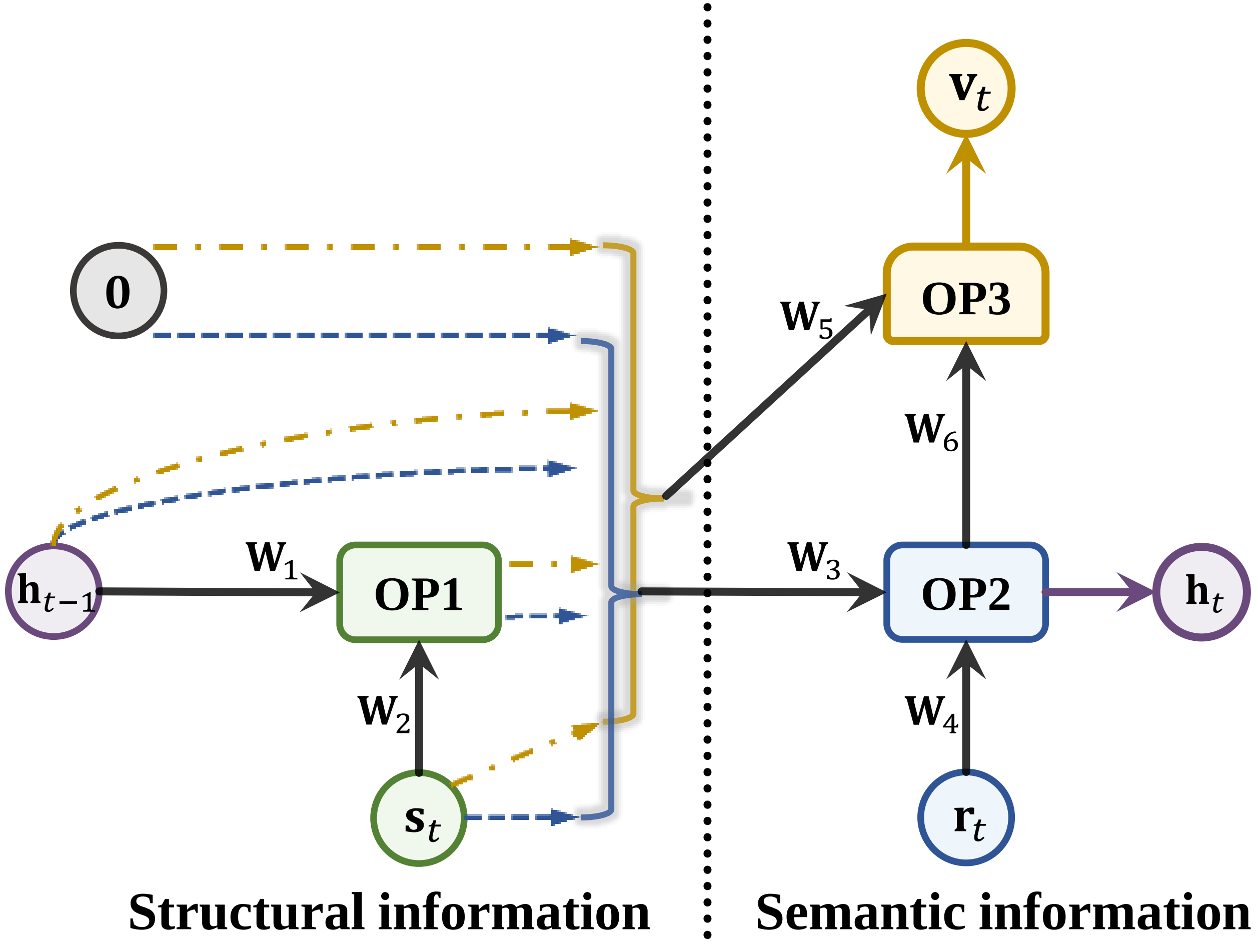
Table 5: Performance in Countries datasets.

	S1	S2	S3
P1	0.998±0.001	0.997±0.002	0.933±0.031
P2	**1.000±0.000**	0.999±0.001	0.952±0.023
P3	0.992±0.001	**1.000±0.000**	0.961±0.016
P4	0.977±0.028	0.984±0.010	**0.964±0.015**
NRASE	**1.000±0.000**	**1.000±0.000**	**0.968± 0.007**

Search space

Distiller: $[\mathbf{v}_t, \mathbf{h}_t] = f(\mathbf{s}_t, \mathbf{r}_t, \mathbf{h}_{t-1})$

combinator	$+, \odot, \otimes$, gate
activation	identity, tanh, sigmoid, relu

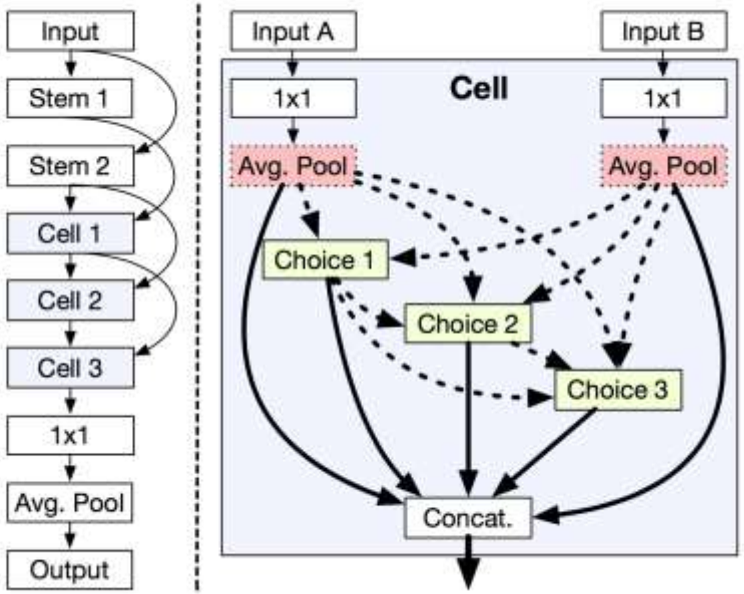


DEFINITION 3 (NAS PROBLEM). *Let the training set be $\mathcal{G}_{tra}$ and validation set be $\mathcal{G}_{val}$. $F(\boldsymbol{\alpha})$ returns the embeddings trained on $\mathcal{G}_{tra}$ with f, of which the architecture is $\boldsymbol{\alpha}$. $\mathcal{M}(F(\boldsymbol{\alpha}), \mathcal{G}_{val})$ measure the performance of embeddings on $\mathcal{G}_{val}$. The problem is to find an architecture $\boldsymbol{\alpha}$ for the path distiller such that validation performance is maximized, i.e.,*

$$\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha} \in \mathcal{A}} \mathcal{M}(F(\boldsymbol{\alpha}), \mathcal{G}_{val}), \tag{2}$$

where $\boxed{\mathcal{A} \text{ is the search space of } \boldsymbol{\alpha}}$ (i.e., containing all possible architectures of f).

Neural architecture search



[Bender et. al. ICML 2018]

Evaluation problem (feedback signal)

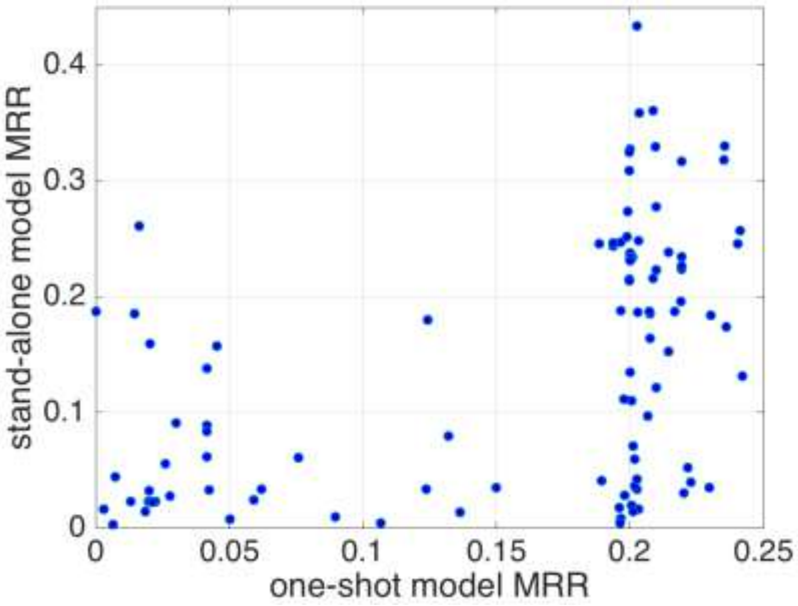
1. Stand-alone: separately train and evaluate (reliable).
2. One-shot: supernet with parameter sharing (efficient).

Gradient problem (optimization direction)

- Gradient information should be obtained from validation measurement.
- But evaluation metric of KGE is non-differentiable.

Search algorithm

- Evaluation problem: for one-shot search, correlations is weak with parameter sharing.



We use stand-alone instead.

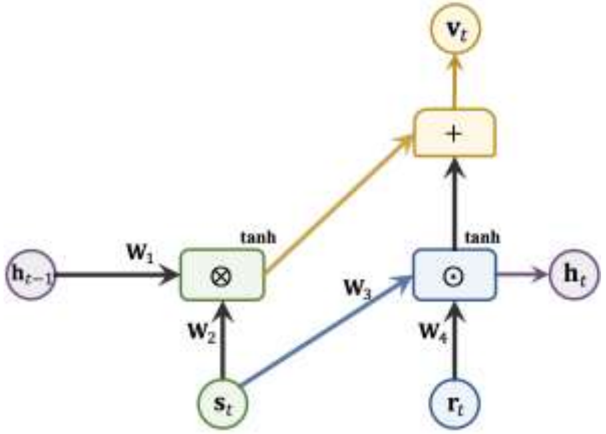
- Gradient problem: refer to derivative-free methods.
  - Natural gradient (NG) descent – a second order method.
  - Stable and have convergence guarantee.

Outline

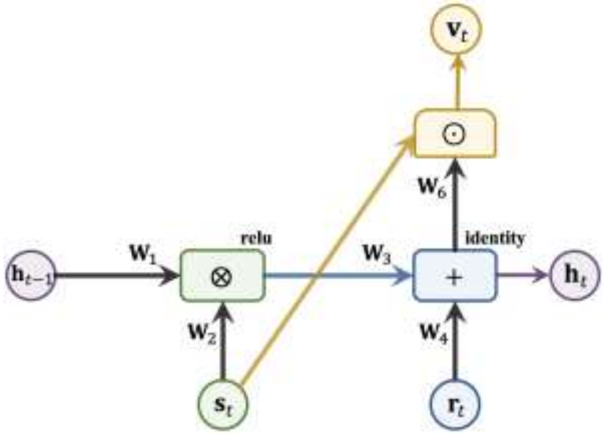
- ➢ **Introduction and Background**
- ➢ **AutoSF: Automated Scoring Function**
- ➢ **NRASE: NAS for Relational Path**
  - Preliminaries
  - Proposed method
  - Empirical results
  - Short summary
- ➢ **Summary**

Effectiveness – entity alignment

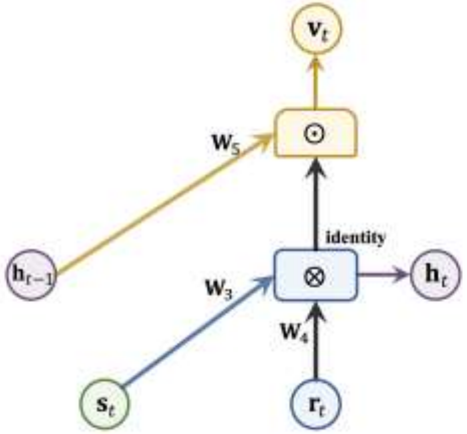
models		DBP-WD			DBP-YG			EN-FR			EN-DE		
		H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
semantic	TransE	28.4	51.4	0.36	27.0	57.4	0.37	16.2	39.0	0.24	40.3	60.9	0.47
	TransD*	27.7	57.2	0.37	17.3	41.6	0.26	21.1	47.9	0.30	24.4	50.0	0.33
	PTransE	16.7	40.2	0.25	7.4	14.7	0.10	7.3	19.7	0.12	27.0	51.8	0.35
structural	BootEA*	32.3	63.1	0.42	31.3	62.5	0.42	31.3	62.9	0.42	44.2	70.1	0.53
	IPTransE*	23.1	51.7	0.33	22.7	50.0	0.32	25.5	55.7	0.36	31.3	59.2	0.41
	GCN-Align*	17.7	27.8	0.25	19.3	41.5	0.27	15.5	34.5	0.22	25.3	46.4	0.33
	ChainR	32.2	60.0	0.42	35.3	64.0	0.45	31.4	60.1	0.41	41.3	68.9	0.51
	RSN*	38.8	65.7	0.49	40.0	67.5	0.50	34.7	63.1	0.44	48.7	72.0	0.57
NRASE (proposed)		**40.7**	**71.2**	**0.51**	**40.2**	**72.0**	**0.51**	**35.5**	**67.9**	**0.46**	**50.2**	**74.9**	**0.59**



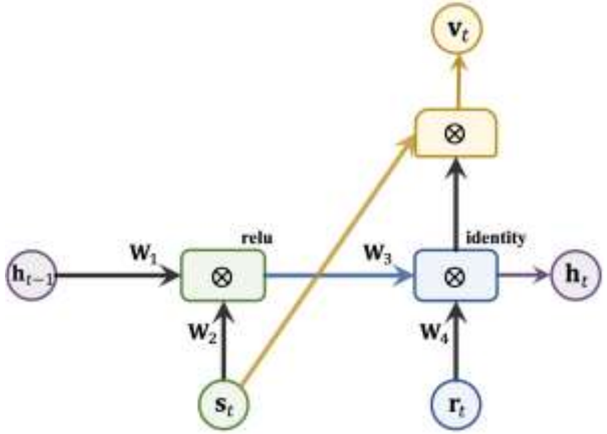
DBP-WD



DBP-YG



EN-FR



EN-DE

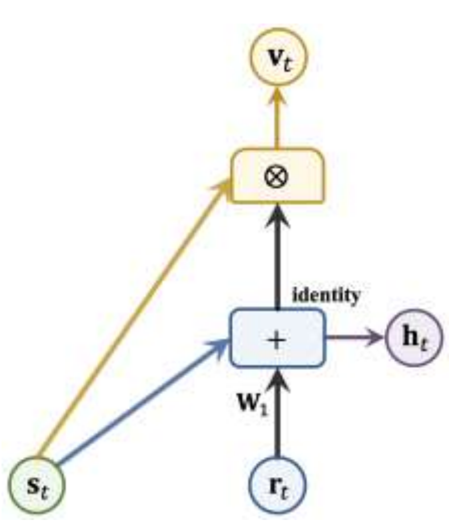
Effectiveness – link prediction

Table 8: Performance comparison on link prediction tasks.

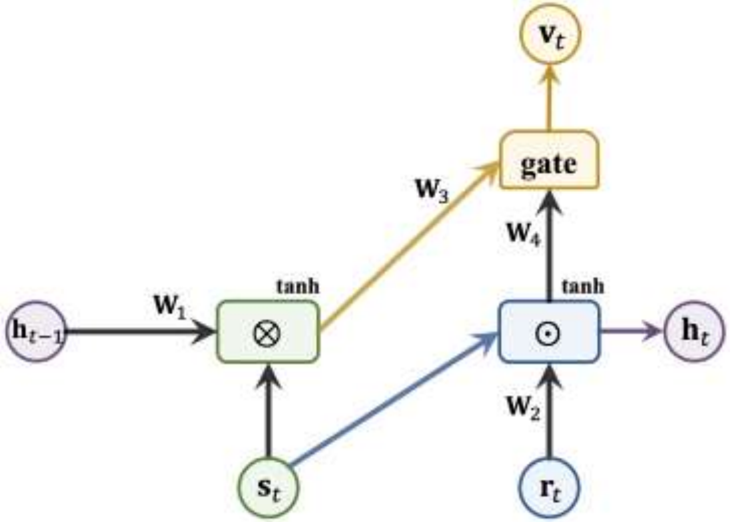
models	WN18-RR			FB15k-237		
	H@1	H@10	MRR	H@1	H@10	MRR
TransE	12.5	44.5	0.18	17.3	37.9	0.24
ComplEx	41.4	49.0	0.44	22.7	49.5	0.31
RotatE	43.6	54.2	0.47	**23.3**	50.4	**0.32**
PTransE	27.2	46.4	0.34	20.3	45.1	0.29
ChainR	28.1	37.9	0.32	21.9	44.4	0.29
RSN	38.0	44.8	0.40	19.2	41.8	0.27
NRASE	**44.0**	**54.8**	**0.48**	**23.3**	**50.8**	**0.32**

Table 12: Percentage of the n-hop triplets in validation and testing datasets.

Datasets		Hops			
		≤ 1	2	3	≥ 4
WN18-RR	validation	35.5%	8.8%	22.2%	33.5%
	testing	35.0%	9.3%	21.4%	34.3%
FB15k-237	validation	0%	73.2%	26.1%	0.7%
	testing	0%	73.4%	26.8%	0.8%

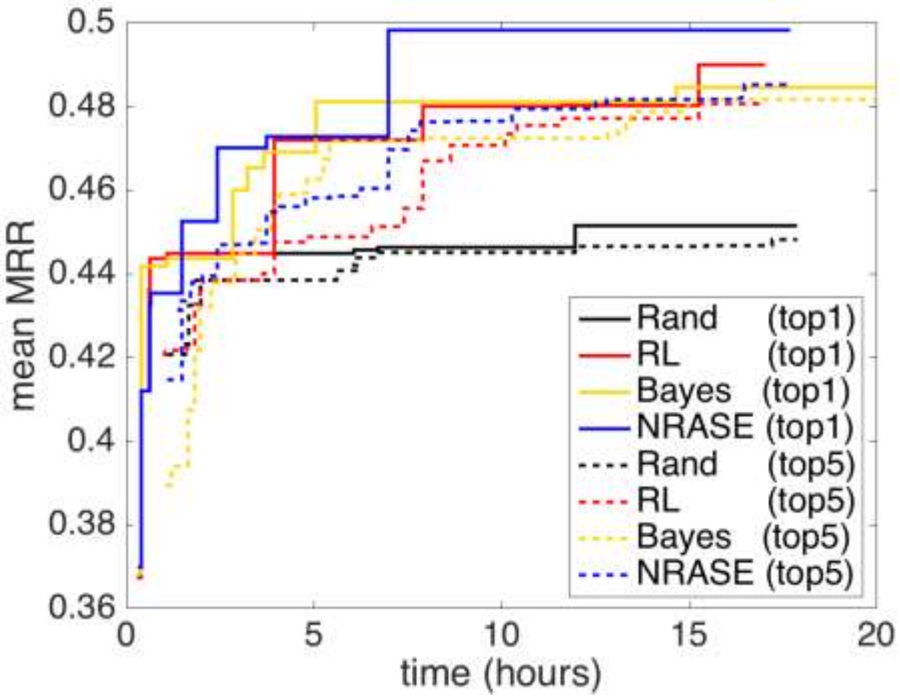


WN18-RR

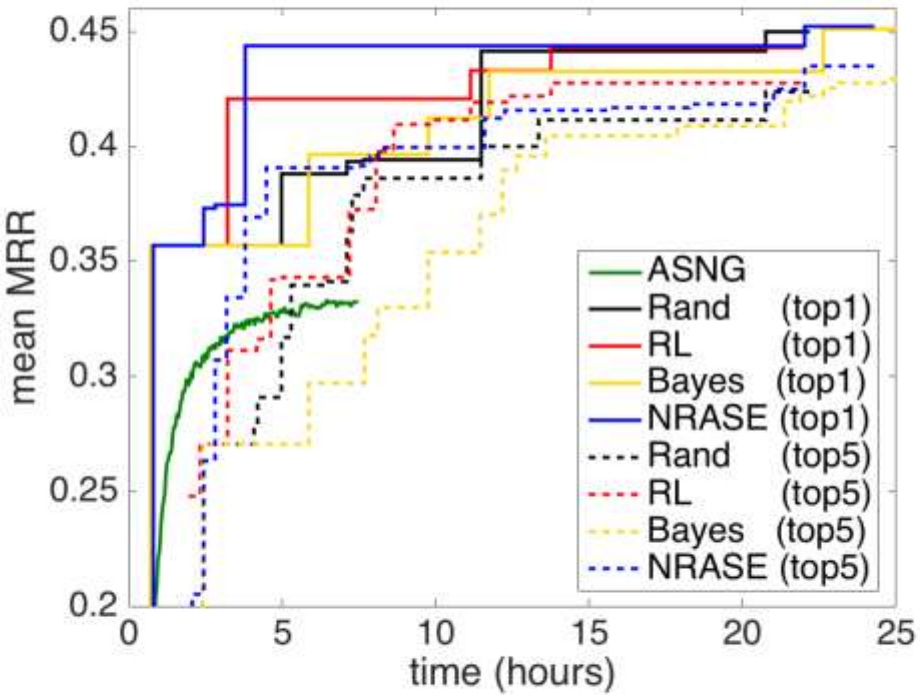


FB15k-237

Efficiency of the hybrid search algorithm



Entity alignment



Link prediction

Outline

- **Introduction and Background**
- **AutoSF: Automated Scoring Function**
- **NRASE: NAS for Relational Path**
  - Preliminaries
  - Proposed method
  - Empirical results
  - Short summary
- **Summary**

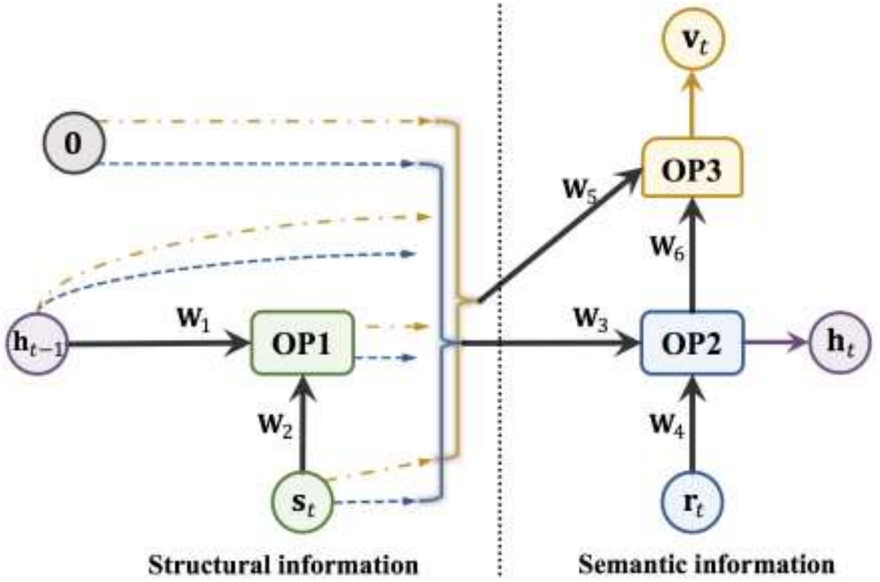
Summary of NRASE

Challenges:

- When and how to leverage structural information is task and data dependent.

Ours:

- Explored the difficulty and importance of processing structural and semantic information in KG.
- Proposed a domain-specific search space for RNN and use Natural Gradient based search algorithm to search efficiently.



- Neural Recurrent Structure Search for Knowledge Graph Embedding. **Y. Zhang**, Q. Yao, L. Chen. *arXiv:1911.07132*

Outline

- ➢ **Introduction and Background**
- ➢ **AutoSF: Automated Scoring Function**
- ➢ **NRASE: NAS for Relational Path**
  - Preliminaries
  - Proposed method
  - Empirical results
  - Short summary
- ➢ **Summary**

Summary

AutoKGE

KGE Problems	Our work	Key idea	AutoML Techniques
Scoring function	AutoSF	Task-aware scoring function	Greedy Search + Domain Property
Relational path	NRASE	Design network to process paths	Hybrid Neural Architecture Search

Thank you!

Q & A

Code is available in https://github.com/AutoML-4Paradigm