AutoSF: Searching Scoring Functions for Knowledge Graph Embedding

Yongqi Zhang[❄], Quanming Yao[♈], Wenyuan Dai[♈], Lei Chen[❄]

[❄]Hong Kong University of Science and Technology
[♈]4Paradigm Inc.

[❄]{yzhangee, leichen}@cse.ust.hk, [♈]{yaoquanming, daiwenyuan}@4paradigm.com

Outline

- **Introduction**
  - Introduction to KG Embedding
  - Introduction to AutoML
- **Proposed method - AutoSF**
  - Problem definition
  - Search space and algorithm
- **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]

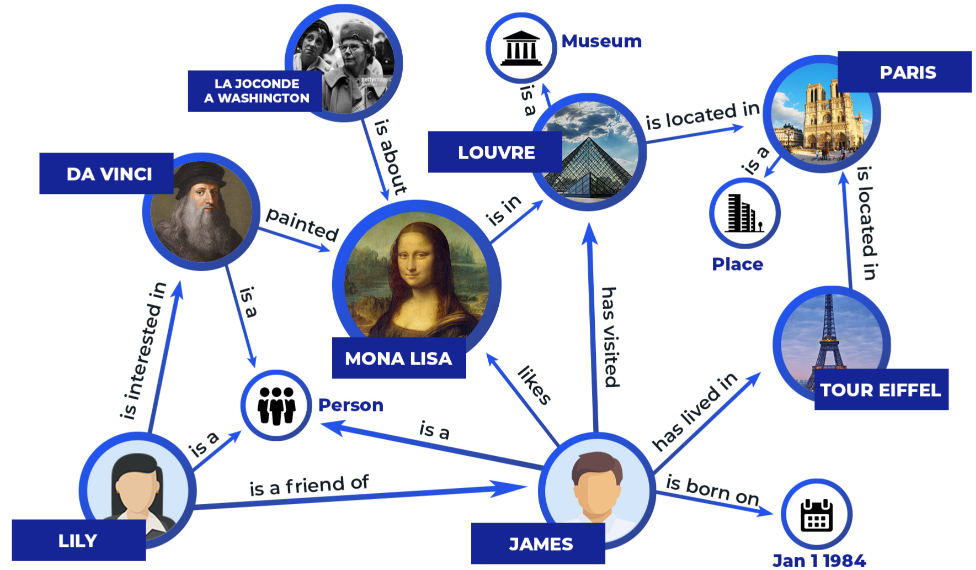
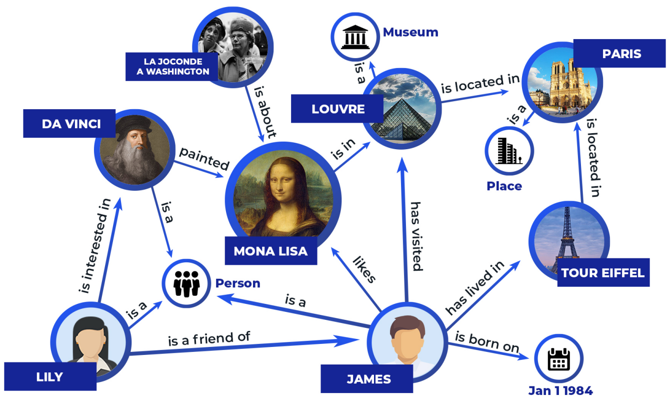


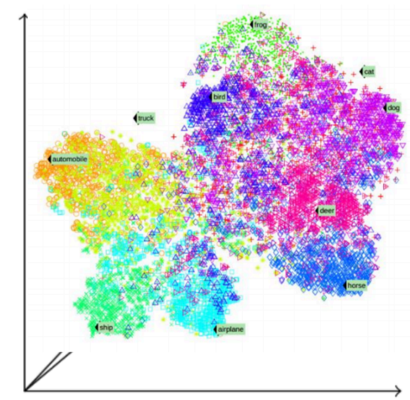
Fig. from [Yashu Seth, 2019]

KG embedding

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

Advantages:

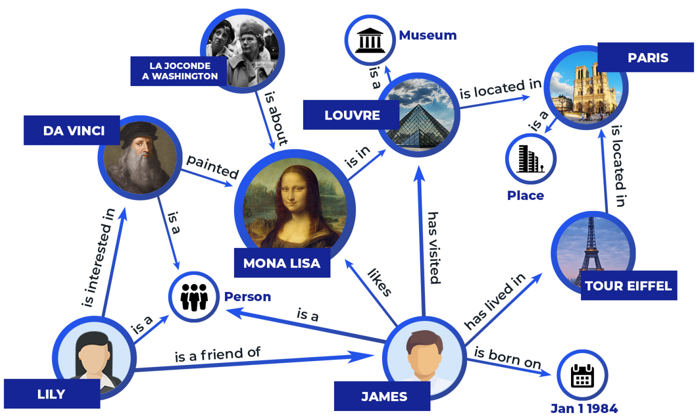
- Inject into downstream ML pipelines.
- Provide efficient similarity search.
- Discover latent properties in missing links.

Learning framework

- Objectives:

$$\max_{\boldsymbol{w}} \overbrace{f^{+}(\underbrace{\boldsymbol{w}; S^{+}) + f^{-}(\boldsymbol{w}}_{\text{parameters}}; S^{-})}^{\text{model}}$$

iterative optimization → Improve performance



Observed triplet S^{+}:
increase score

Unobserved triplet S^{-}:
decrease score

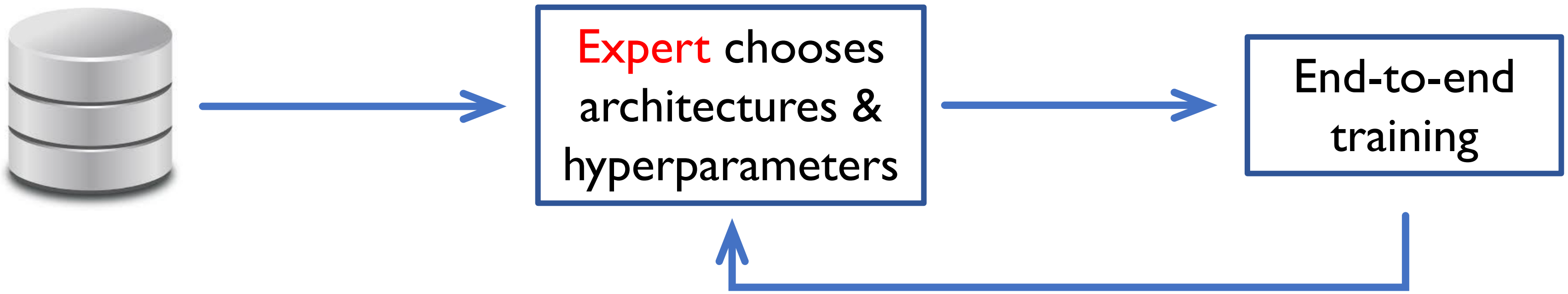
Triplet with higher score is more likely to be positive. → Predict missing links.

Outline

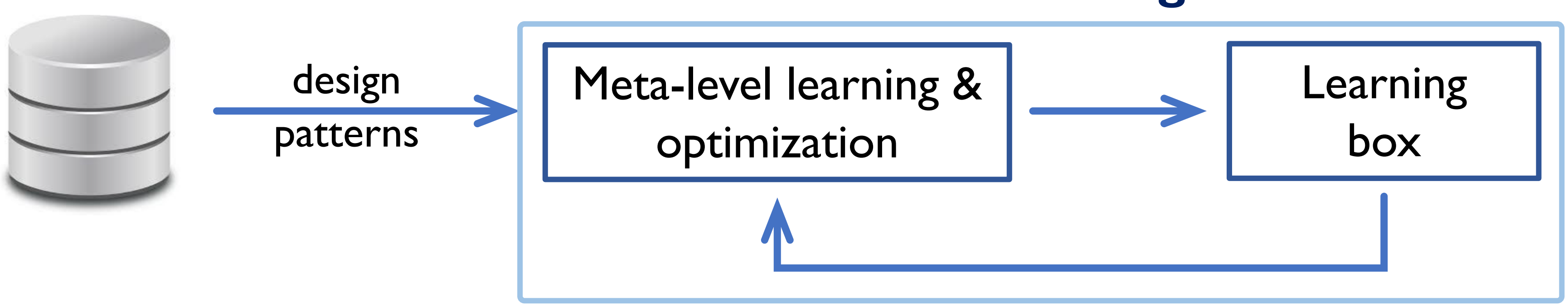
- **Introduction**
  - Introduction to KG Embedding
  - Introduction to AutoML
- **Proposed method - AutoSF**
  - Problem definition
  - Search space and algorithm
- **Experiments**
- **Summary**

Automated machine learning

General deep learning practice



AutoML: true end-to-end learning



Search space: what to be searched

- hyper-parameters, neural network structures.

Search algorithm: how to search efficiently

- Reinforcement learning, Bayes optimization, evolution algorithm.

Outline

- **Introduction**
  - Introduction to KG Embedding
  - Introduction to AutoML
- **Proposed method - AutoSF**
  - Problem definition
  - Search space and algorithm
- **Experiments**
- **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.

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]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$-\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{1/2}$	$\|\mathbf{h}\|_2 = 1, \|\mathbf{t}\|_2 = 1$
TransH [15]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$			
TransR [16]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$			
TransD [50]	$\mathbf{h}, \mathbf{w}_h \in \mathbb{R}$ $\mathbf{t}, \mathbf{w}_t \in \mathbb{R}^d$			
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]	$\mathbf{h} \sim \mathcal{N}(\boldsymbol{\mu}_h$ $\mathbf{t} \sim \mathcal{N}(\boldsymbol{\mu}_t,$ $\boldsymbol{\mu}_h, \boldsymbol{\mu}_t \in \mathbb{R}$ $\Sigma_h, \Sigma_t \in \mathbb{R}$			
TransG [46]	$\mathbf{h} \sim \mathcal{N}(\boldsymbol{\mu}_h$ $\mathbf{t} \sim \mathcal{N}(\boldsymbol{\mu}_t,$ $\boldsymbol{\mu}_h, \boldsymbol{\mu}_t \in \mathbb{R}$			
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)

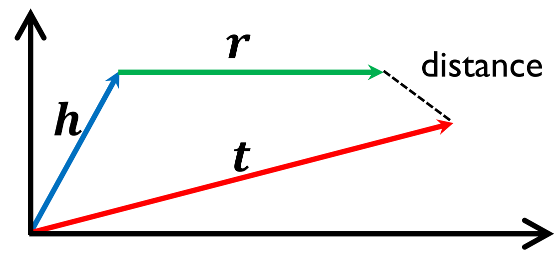
Method	Ent. embedding	Rel. embedding	Scoring function $f_r(h,t)$	Constraints/Regularization
RESCAL [13]	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{M}_r \in \mathbb{R}^{d\times d}$	$\mathbf{h}^\top \mathbf{M}_r \mathbf{t}$	$\|\mathbf{h}\|_2 \le 1, \|\mathbf{t}\|_2 \le 1, \|\mathbf{M}_r\|_F \le 1$ $\mathbf{M}_r = \sum_i \pi_r^i \mathbf{u}_i \mathbf{v}_i^\top$ (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]

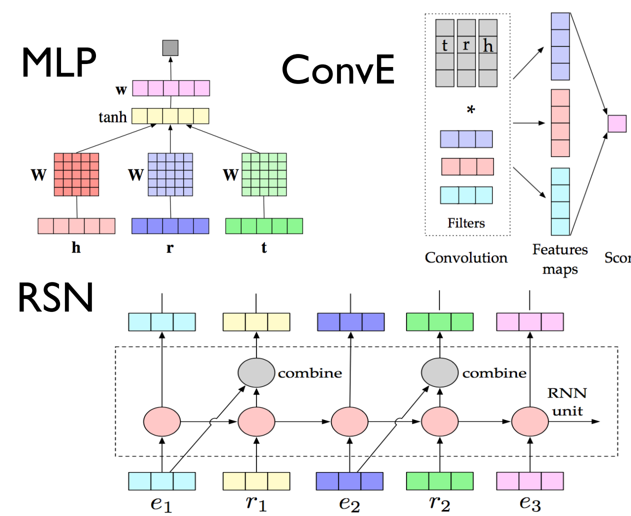
- Design principles:
  - Encode entity and relation into some space to measure the plausibility.
  - Capture important properties:
    - symmetric, anti-symmetric, inverse, asymmetric…

General types

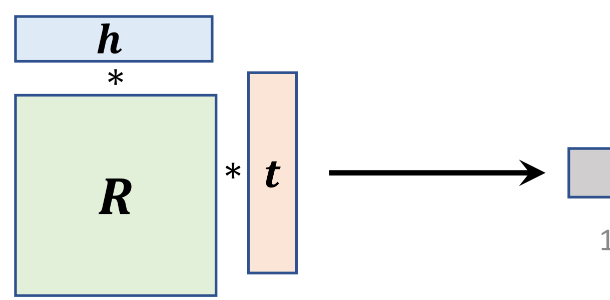
- Translation Distance Models (TDMs)
  - TransE, TransH, RotatE, etc.
  - less expressive. [Wang et. al. AAAI 2017]



- 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]



Bilinear models

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

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

symmetric	√
anti-symmetric	×
asymmetric	×
inverse	×

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

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

symmetric	√
anti-symmetric	√
asymmetric	√
inverse	√

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

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

symmetric	√
anti-symmetric	√
asymmetric	√
inverse	√

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

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

symmetric	√
anti-symmetric	√
asymmetric	√
inverse	√

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.

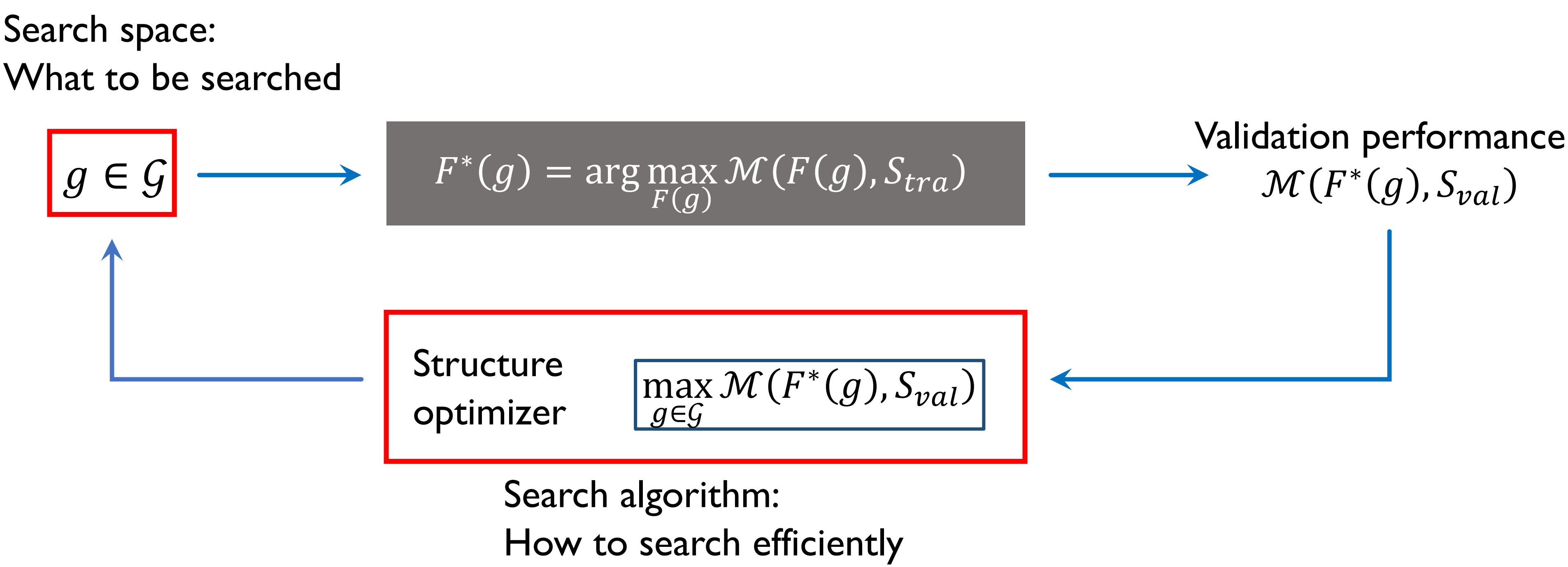
AutoSF: Definition

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:*

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

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

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



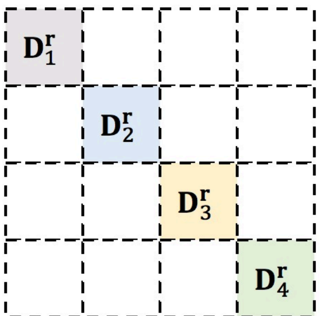
Outline

- **Introduction**
  - Introduction to KG Embedding
  - Introduction to AutoML
- **Proposed method - AutoSF**
  - Problem definition
  - Search space and algorithm
- **Experiments**
- **Summary**

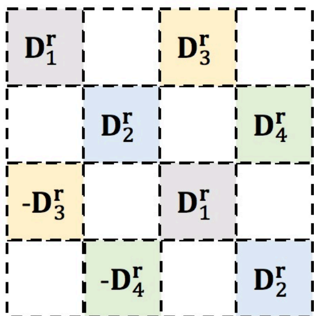
Search space

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 $f_{\mathit{unified}}(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \sum_{i,j} \langle \mathbf{h}_i, \mathbf{a}_{ij}, \mathbf{t}_j \rangle = \mathbf{h}^\top g(\mathbf{r})\, \mathbf{t}$.*

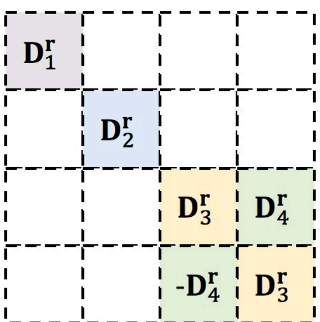
The location of a block matrix $\mathbf{D}_i^r$ represents a multiplicative term.



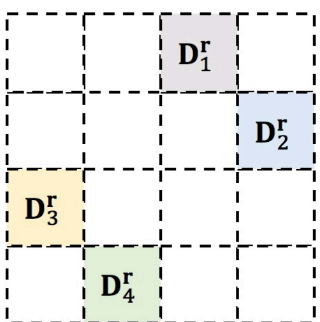
DistMult



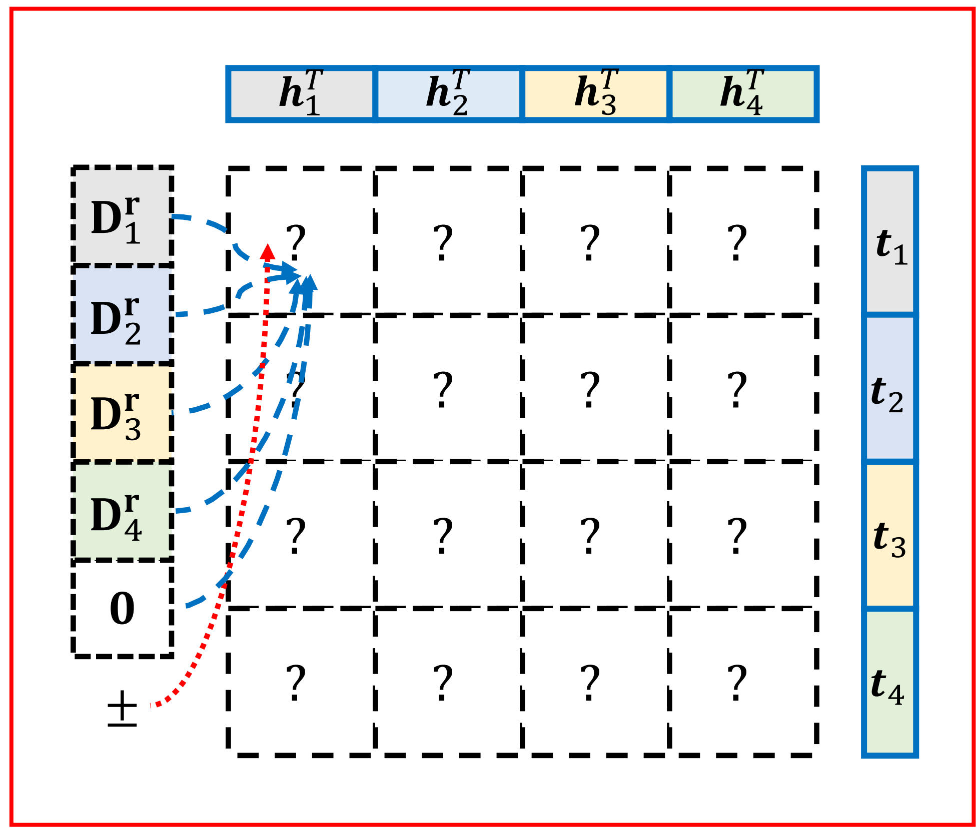
ComplEx



Analogy

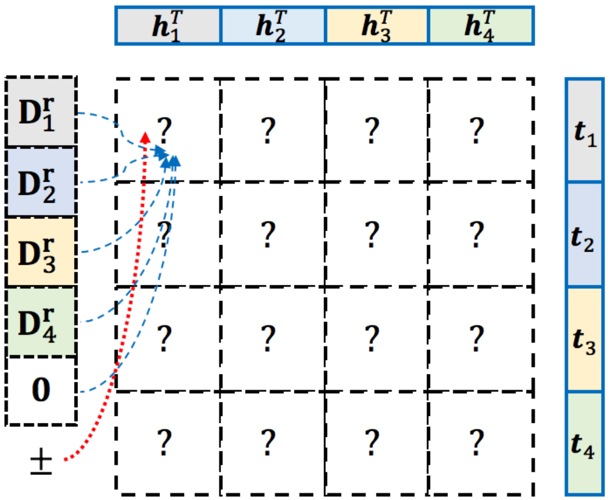


SimplE



Challenges

1. Size of search space is very large: 9^{16}.
2. Cost of training and evaluating a specific model structure is expensive.
3. How to capture important properties like symmetric, asymmetric?

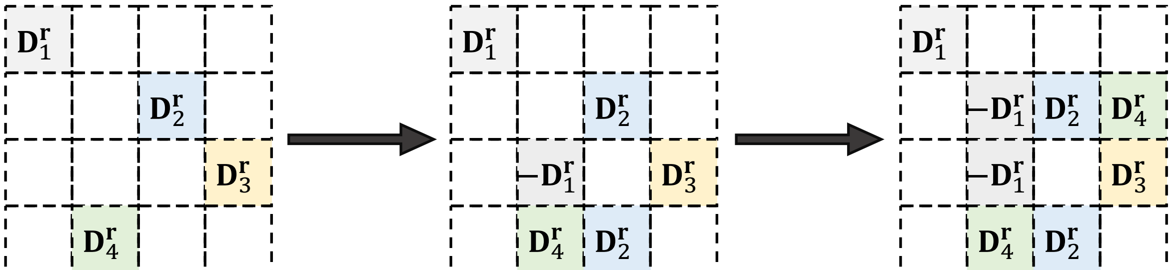


Key point: not all scoring functions / structures need to be trained.

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

Search algorithm

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



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

For f^4, reduces from 9216 to 5.

➢ Filter: remove bad and equivalent SFs.

- Bad: there are zero/repeated rows/columns.
- Equivalent: have the same expressive ability after permutation or slipping signs.

➢ 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**
  - Introduction to KG Embedding
  - Introduction to AutoML
- **Proposed method - AutoSF**
  - Problem definition
  - Search space and algorithm
- **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';

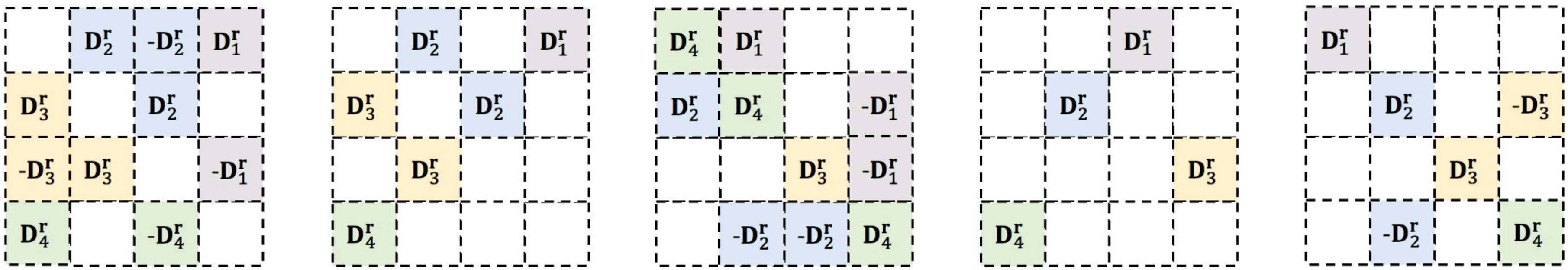
Metrics

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

		WN18			FB15k			WN18RR			FB15k237			YAGO3-10		
type	model	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
TDM	TransE [54]	0.500	—	94.1	0.495	—	77.4	0.178	—	45.1	0.256	—	41.9	—	—	—
	TransH [54]	0.521	—	94.5	0.452	—	76.6	0.186	—	45.1	0.233	—	40.1	—	—	—
	RotatE [35]	0.949	94.4	95.9	0.797	74.6	88.4	0.476	42.8	**57.1**	0.338	24.1	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]	0.942	93.5	95.5	0.745	67.0	87.3	0.46	39.	48.	0.316	23.9	49.1	0.52	45.	66.
BLM	TuckER [1]	**0.953**	**94.9**	95.8	0.795	74.1	89.2	0.470	44.3	52.6	0.358	26.6	54.4	—	—	—
	HolEX [45]	0.938	93.0	94.9	0.800	75.0	88.6	—	—	—	—	—	—	—	—	—
	QuatE [53]	0.950	94.5	95.9	0.782	71.1	90.0	0.488	43.8	**58.2**	0.348	24.8	55.0	—	—	—
	DistMult	0.821	71.7	95.2	0.817	77.7	89.5	0.443	40.4	50.7	0.349	25.7	53.7	0.552	47.6	69.4
	ComplEx	0.951	94.5	95.7	0.831	79.6	90.5	0.471	43.0	55.1	0.347	25.4	54.1	0.566	49.1	70.9
	Analogy	0.950	94.6	95.7	0.829	79.3	90.5	0.472	43.3	55.8	0.348	25.6	54.7	0.565	49.0	71.3
	SimplE/CP	0.950	94.5	95.9	0.830	79.8	90.3	0.468	42.9	55.2	0.350	26.0	54.4	0.565	49.1	71.0
AnyBURL [27]		0.95	94.6	95.9	0.83	80.8	87.6	0.48	44.6	55.5	0.31	23.3	48.6	0.54	47.7	47.3
AutoSF		0.952	94.7	**96.1**	**0.853**	**82.1**	**91.0**	**0.490**	**45.1**	56.7	**0.360**	**26.7**	**55.2**	**0.571**	**50.1**	**71.5**

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

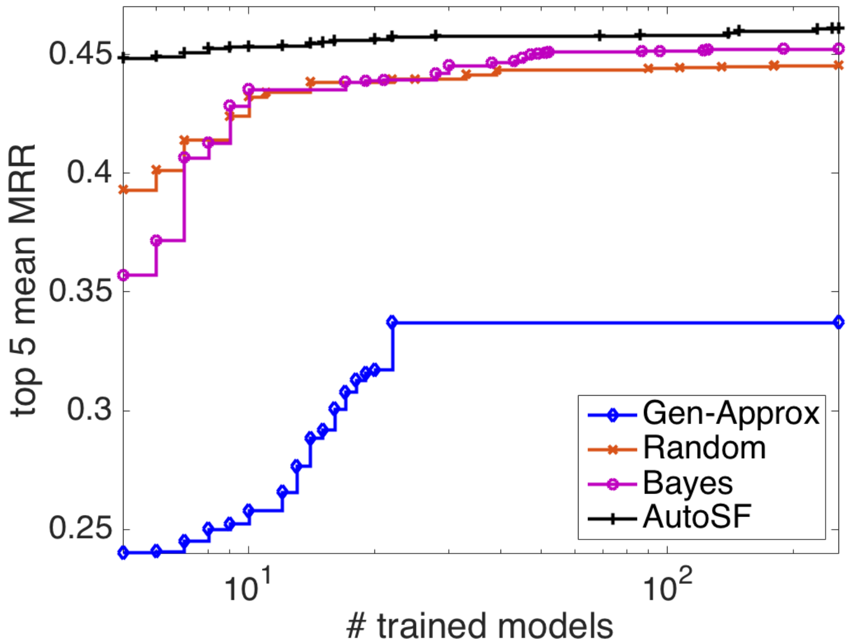
Distinctiveness



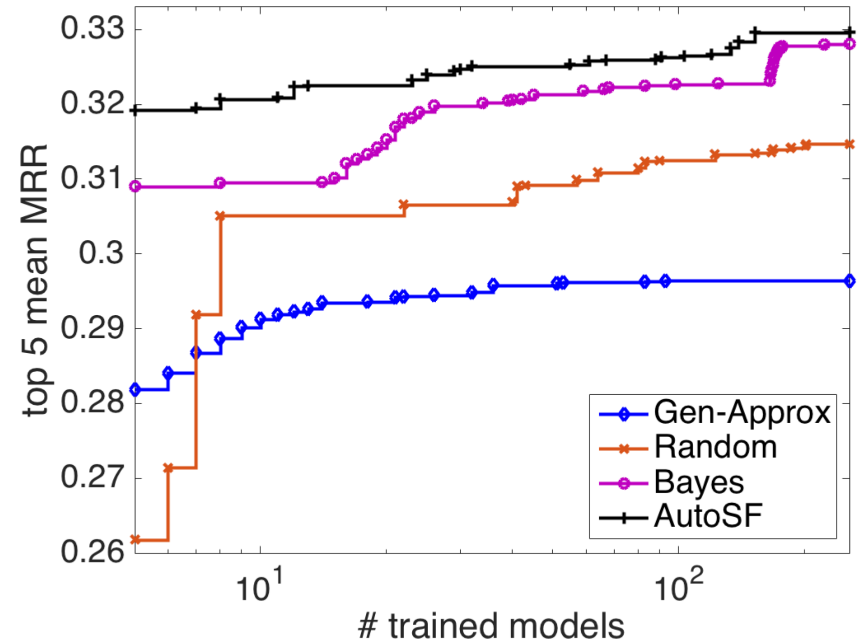
(a) WN18. (b) FB15k. (c) WN18RR. (d) FB15k237. (e) YAGO3-10.

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

Efficiency



WN18-RR



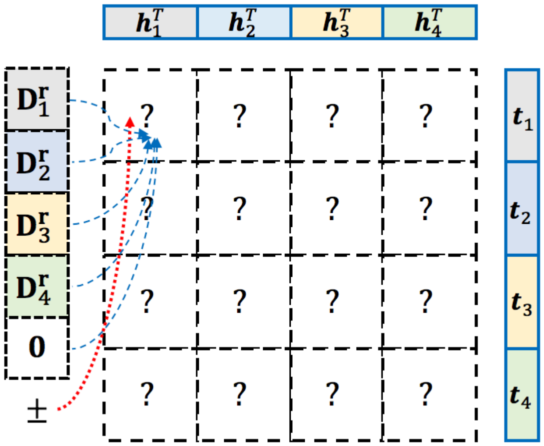
FB15k237

- 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**
  - Introduction to KG Embedding
  - Introduction to AutoML
- **Proposed method - AutoSF**
  - Problem definition
  - Search space and algorithm
- **Experiments**
- **Summary**

Summary



Challenges:

- Designing new and universal SFs are non-trivial.
- Different KG has distinct properties.
- How to design domain specific search space and efficient search algorithm?

Contributions:

- The first AutoML approach for KGE to learn task-aware SFs.
- Well-defined search space and search algorithm with domain knowledge.

Future work:

- Search space beyond bilinear models.
- Enhance search efficiency.

Thank You

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

Open Positions: Intern and full-time opportunities for Machine Learning Research@4Paradigm.
Please send your CV to yaoquanming@4paradigm.com.