Toward Automated Knowledge Graph Representation Learning

Dr. 姚权铭

第四范式 - 高级科学家

机器学习研究组创建者&负责人

E-mail: yaoquanming@4paradigm.com

2020.11.15

Outline

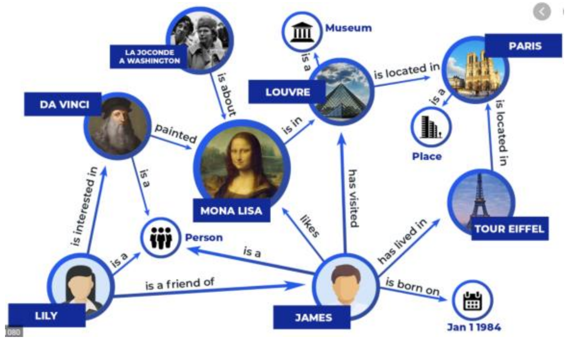
1. What is Knowledge Graph (KG)?
    - Importance & Core issues
2. What is Automated Machine Learning (AutoML)?
3. Attacking Core Issues in KG by AutoML
4. Summary

Knowledge Graph (KG)

A collection of interlinked descriptions of entities – objects, events or concepts

- Connect human understandings with computer computation power

Computer



KG

Human

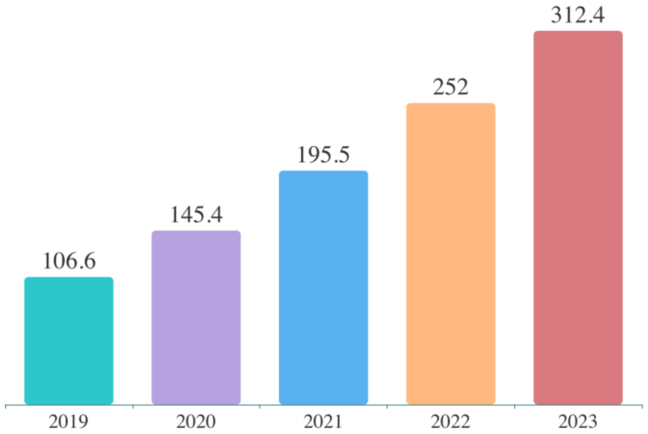
FROM SYSTEM 1 DEEP LEARNING TO SYSTEM 2 DEEP LEARNING

YOSHUA BENGIO

2018 Turing Award

NeurIPS'2019 Posner Lecture
December 11th, 2019, Vancouver BC

Academia: Cognitive Computing



Industry Market

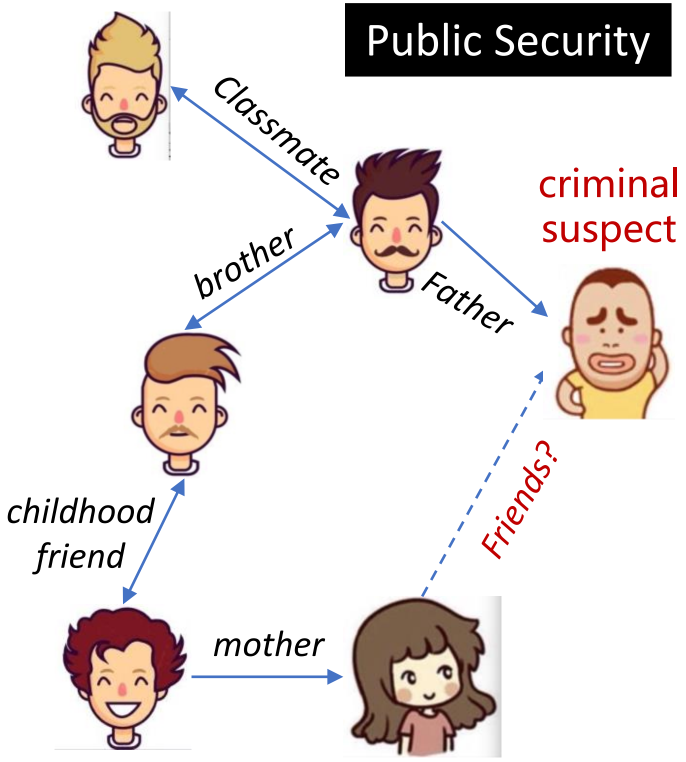
- Exceed 30 billion RMB in 2023
- Annual growth rate of 30.8%

Government Support

1.3 认知计算基础理论与方法研究

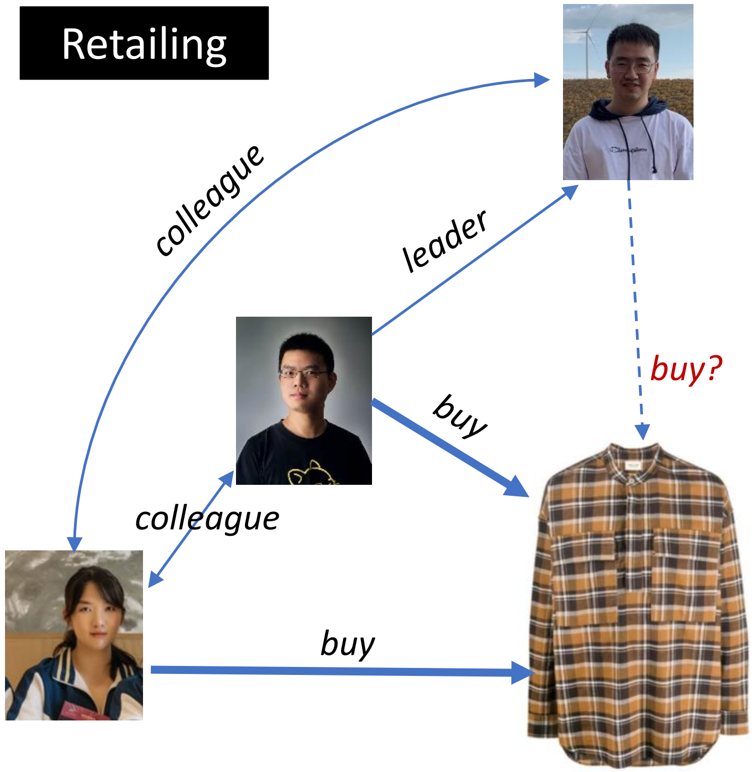
研究内容：聚焦开放、动态、真实环境下推理与决策重大问题，开展常识学习、直觉推理、自主演化、因果分析等理论和方法研究，重点突破刻画环境自适应、不完全推理、自主学习、对抗学习、智能体协同优化等特点的认知计算理论和算法，在跨媒体智能、自主智能、群体智能、人机混合或混合增强智能等智能形态方面实现应用验证。

KG – Application examples



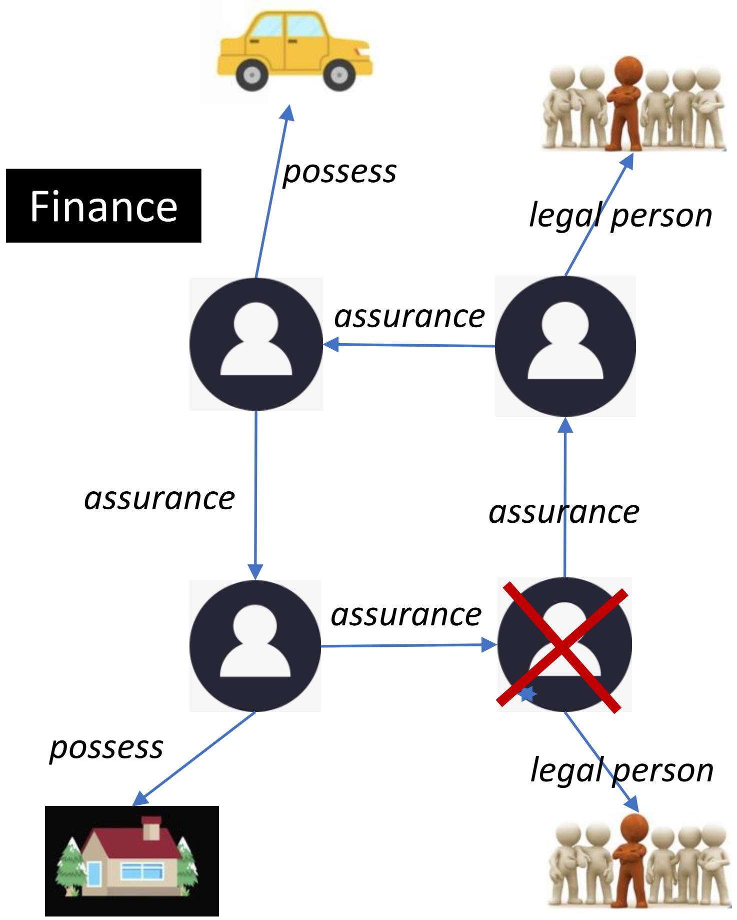
Person of Interest

Find contact



Recommendation

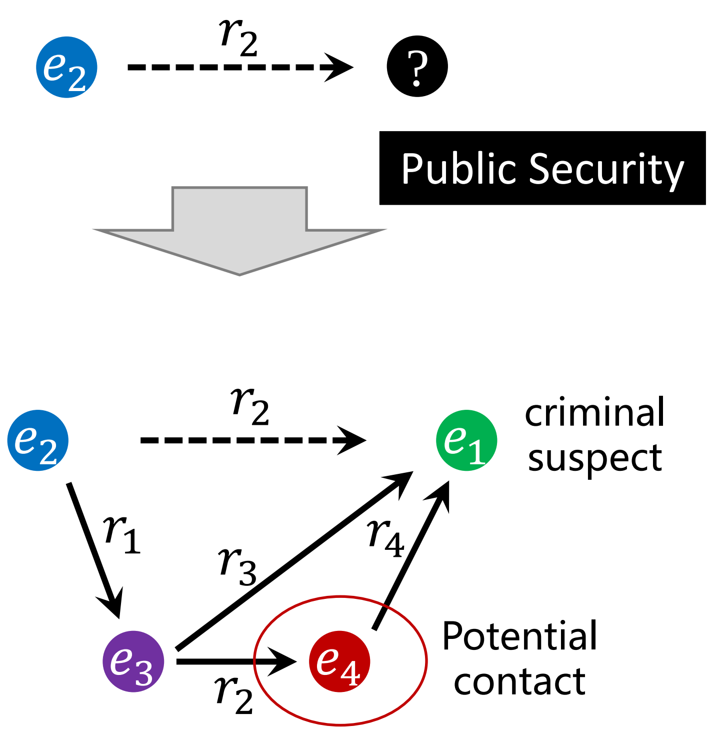
Track preference



Bank Credits

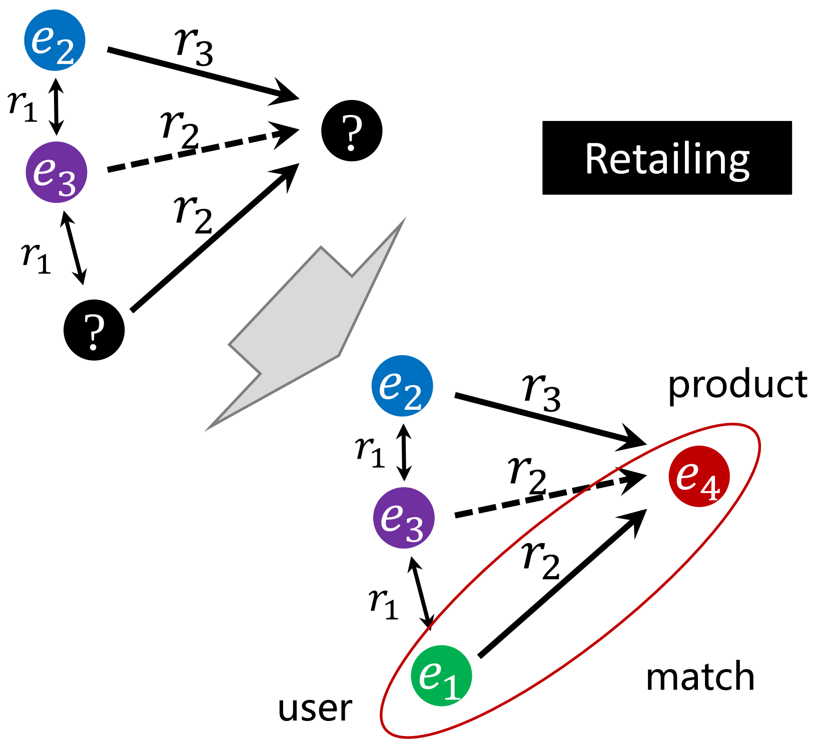
Money chain

KG – Learning tasks



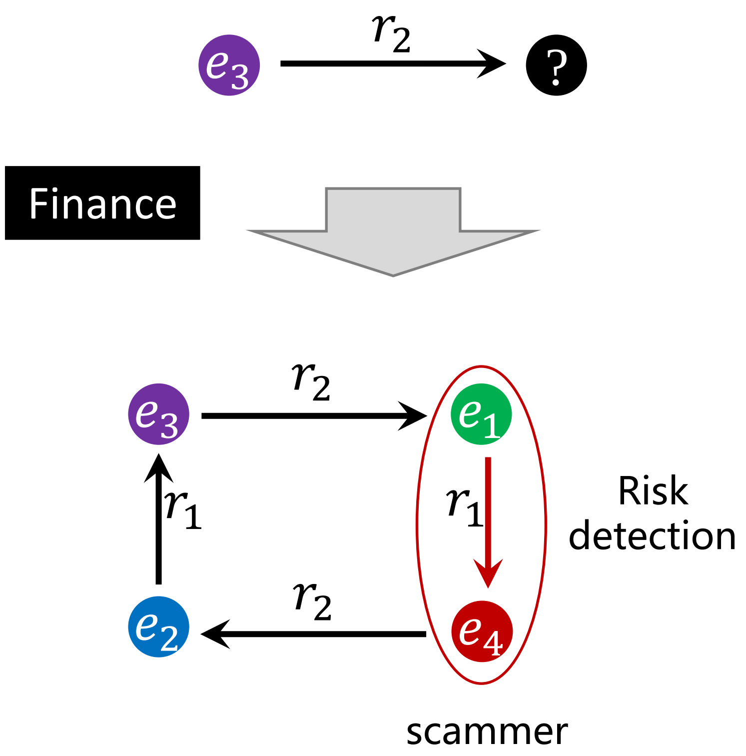
Person of Interest

Find contact



Recommendation

Track preference



Bank Credits

Money chain

KG – Core issues

Knowledge Graph = Knowledge + Graph

Semantics: Symmetric, inverse, asymmetric, composition…

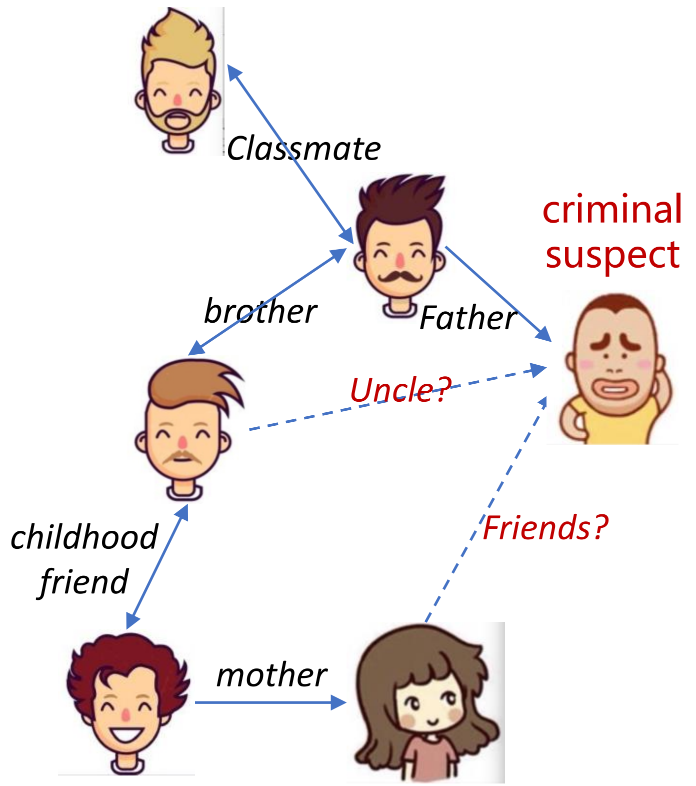
- (A, *isBrotherOf*, B) ∧ (B, *isFatherOf*, C) ⇒ (A, *isUncleOf*, C)
- (A, *spouse*, B) ⟺ (B, *spouse*, A)
- (A, *older*, B) ⟺ (B, *younger*, A)
- (A, *location*, USA)

Topology: A directed multi-relational graph

A graph-structured representation

Whole graph/subgraph as input

How to exploit semantic and topological information?



Outline

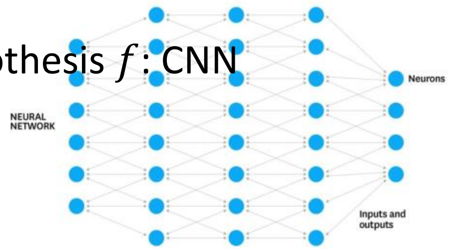
1. What is Knowledge Graph (KG)?
2. What is Automated Machine Learning (AutoML)?
3. Attacking Core Issues in KG by AutoML
4. Summary

What is Machine Learning (ML)?

Image Classification

Optimization

Hypothesis f: CNN



Generalization

Accuracy

Design a **hypothesis (function) f** to perform the learning task

Target	Concept	Issue
Better Performance (more important)	Generalization	What kind of f should we use?
Higher Efficiency	Optimization	How can we find such f?

Not everything can be learnt

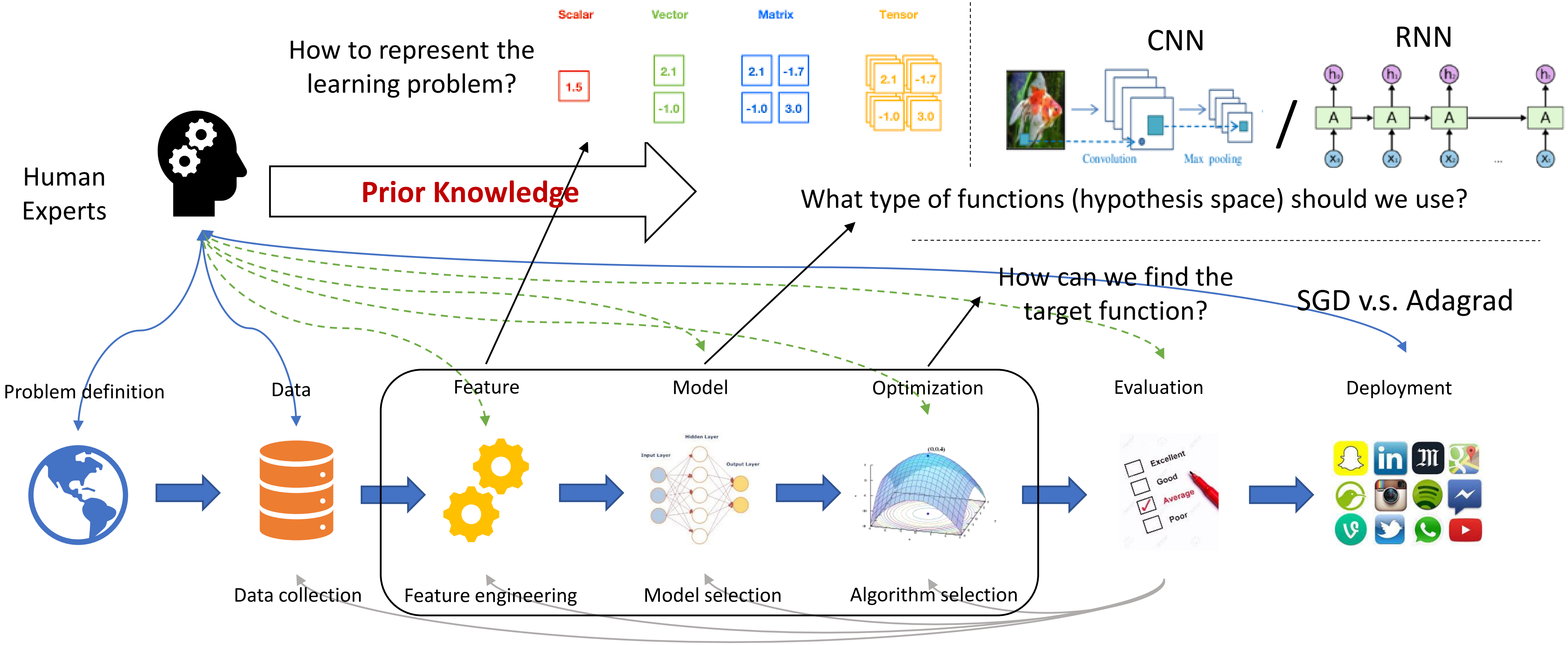
PAC-Learning (Definition 2.3 in [1]): What kind of problems can be solved in polynomial time

No Free Lunch Theorem (Appendix B [2]): No single algorithm can be good on all problems

[1]. M. Mohri, A. Rostamizadeh, A. Talwalkar. Foundations of machine learning. 2018

[2]. O. Bousquet, et.al. Introduction to Statistical Learning Theory. 2016

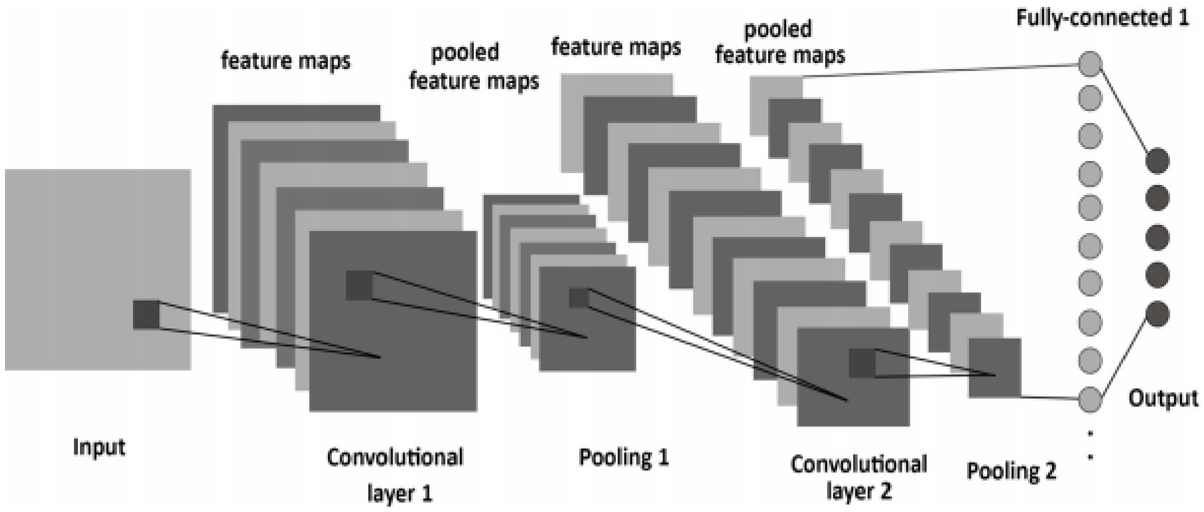
What is Automated ML (AutoML)?



Reduce the usage of humans **(low-level) prior knowledge** in the trial-and-error process of machine learning

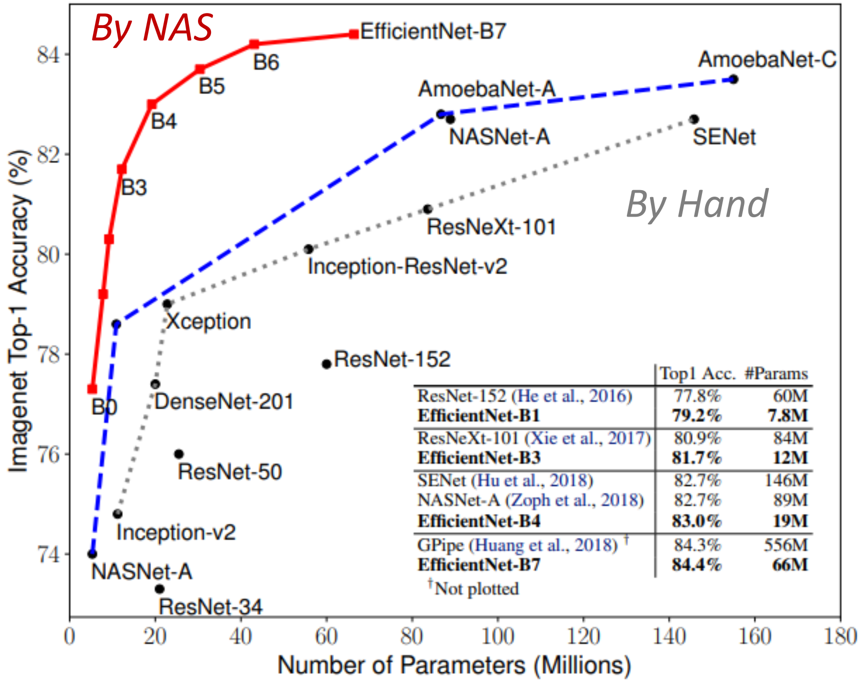
AutoML – Research example

Architecture of networks are critical to deep learning's performance but hard to fine-tune



Design choice in each layer

- number of filters
- filter height
- filter width
- stride height
- stride width
- skip connections



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.2%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.7%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**83.0%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

†Not plotted

Much better than hand-designed ones

Neural Architecture Search (**NAS**) tries to design data-specific architectures

Right figure: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ICML 2019
Left figure: Gradient-based learning applied to document recognition. 1998

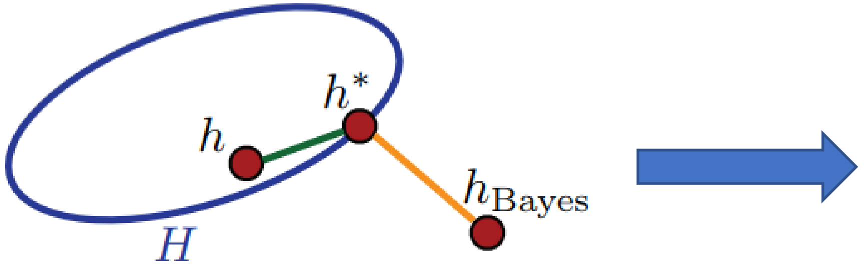
AutoML – Combinatorial generalization

Parameterized the prior knowledge of learning methods, e.g.,

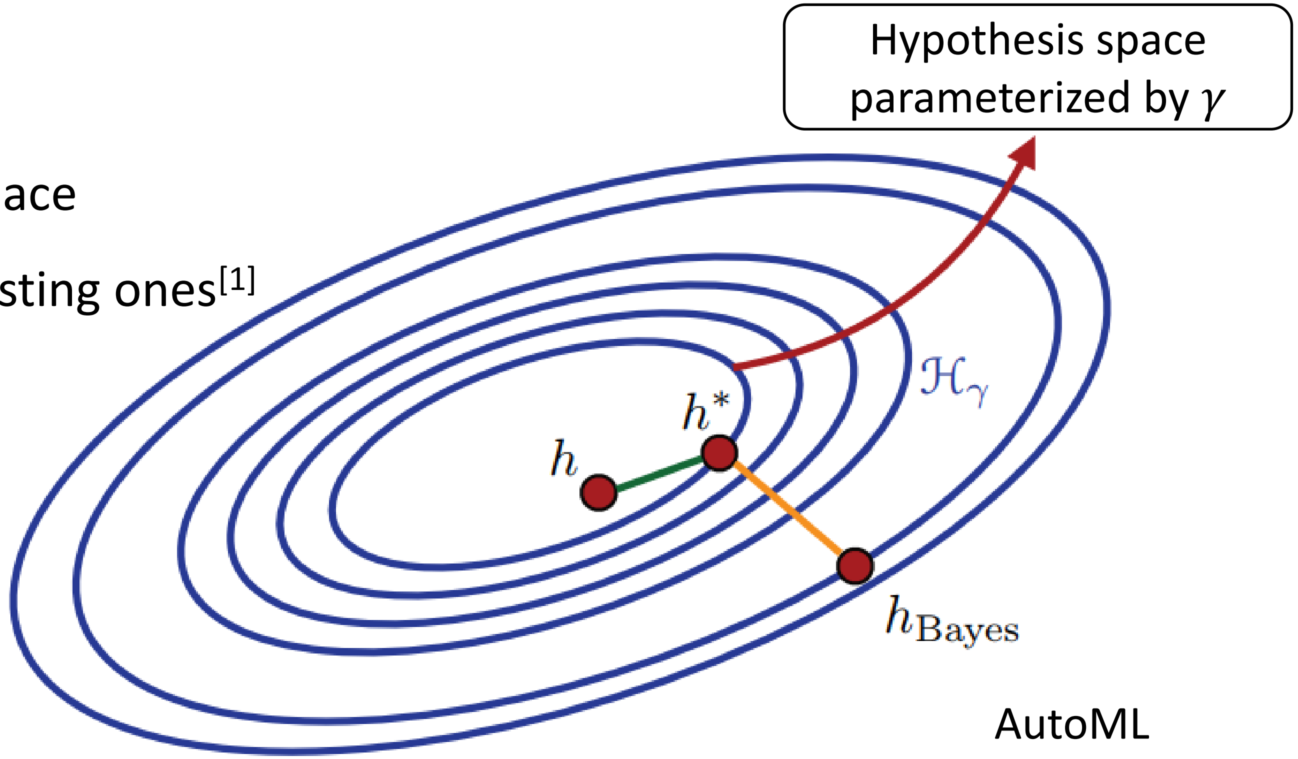
- minimize the total error
- reduce parameter numbers

Perform efficient search in the designed (new) space

- combinatorial generalize new models from existing ones[1]



Classical ML



AutoML

Figure 4.1 & 4.2. M. Mohri, A. Rostamizadeh, A. Talwalkar. Foundations of machine learning. 2018

[1]. P. Battaglia. et.al. Relational inductive biases, deep learning, and graph networks. arXiv 2018.

AutoML – Successor of ML's trend

- Core Issue in Machine Learning: Improving learning performance (with higher efficiency)
- AutoML: an evolving way to improve learning performance

Rule-based	Statistics-based	Deep Learning-based	AutoML-based
Association rules mining	Support vector machine	Convolutional neural networks	Neural architecture search
1970s	1990s	2010s	2017

Better performance →

Continue the trends

- Larger hypothesis (more complex models) are being used
- Optimization is getting complex (even mixed up with generalization)
- The prior knowledge is imposed on more abstract level

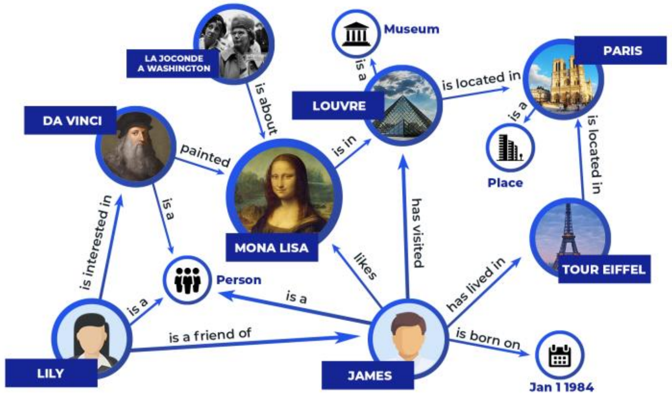
Low-level human prior knowledge is replaced by computation power

Outline

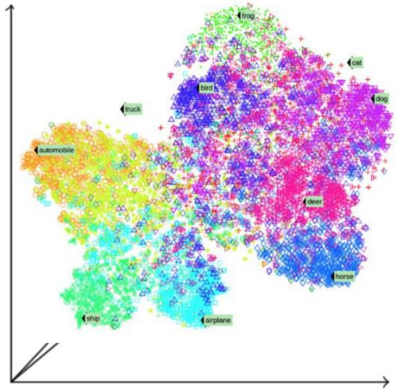
1. What is Knowledge Graph (KG)?
2. What is Automated Machine Learning (AutoML)?
3. Attacking Core Issues in KG by AutoML
    - Overview of Ideas
    - ICDE 2020: Search to Capture Semantics
    - NeurIPS 2020: Search to Exploit Graph Topology
    - Future Works
4. Summary

KG Representation Learning

Encode entities and relations in KG into low-dimensional vector spaces, while capturing nodes' and edges' connection & semantic properties.



input



output

Advantages:

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

Scoring functions (SFs) $f(\mathbf{h}, \mathbf{r}, \mathbf{t})$:

- measure the plausibility of triplets $\{(h, r, t)\}$ in KG.

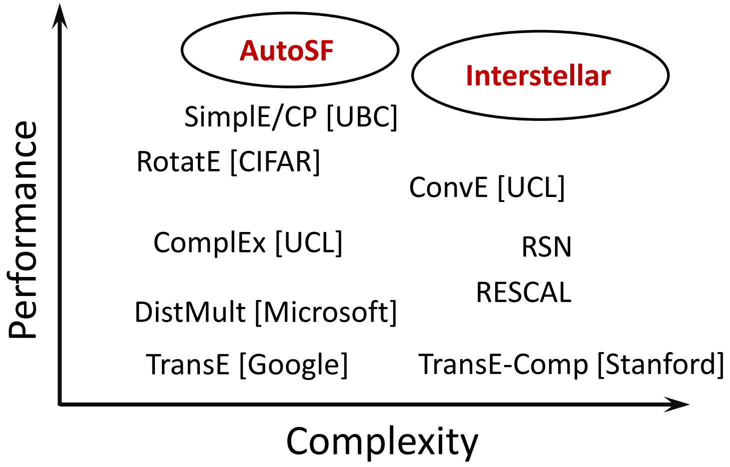
Observed triplet S^+: increase score

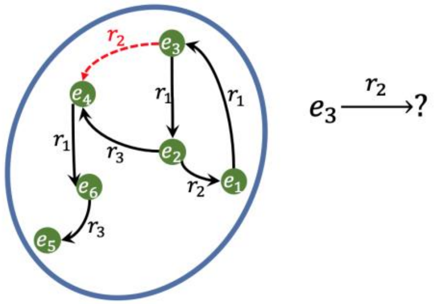
Unobserved triplet S^-: decrease score

Our work – Overview

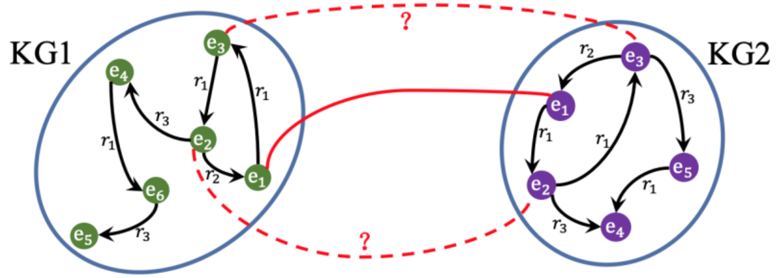
Using AutoML techniques to design data-specific KG learning methods.

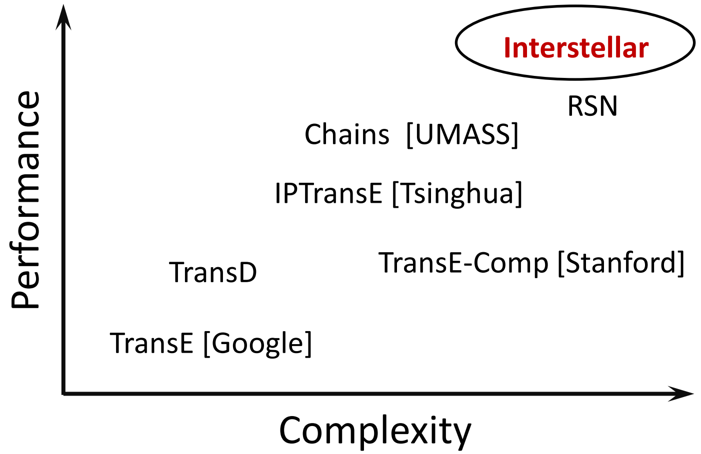
- AutoSF (ICDE 2020): Search to capture semantics
- Interstellar(NeurIPS 2020): Search to exploit graph topology





Link Prediction





Entity Alignment

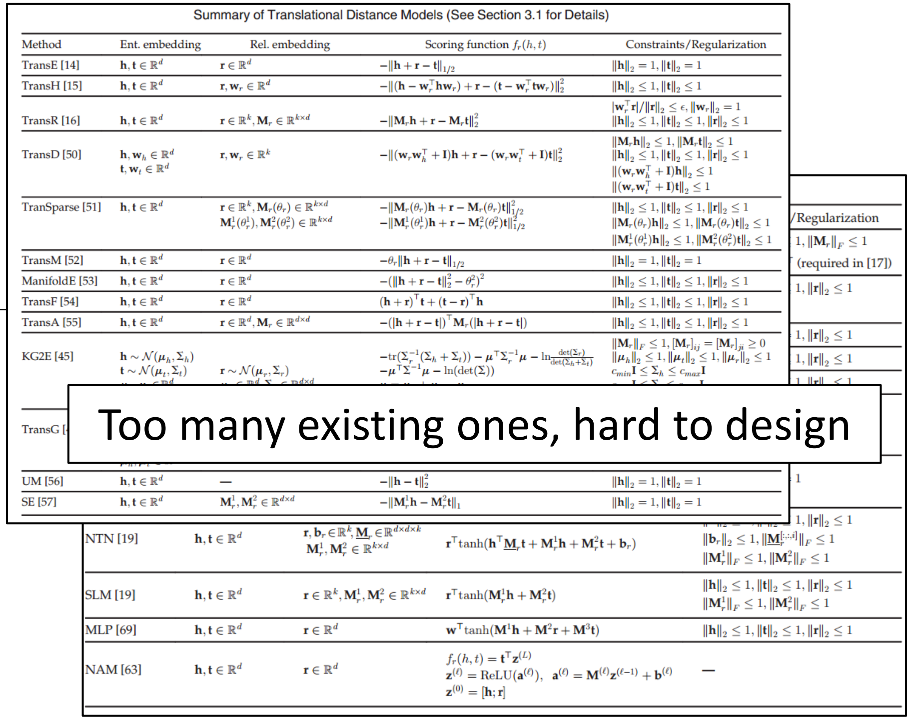
Outline

1. What is Knowledge Graph (KG)?
2. What is Automated Machine Learning (AutoML)?
3. Attacking Core Issues in KG by AutoML
    - Overview of Ideas
    - ICDE 2020: Search to Capture Semantics
    - NeurIPS 2020: Search to Exploit Graph Topology
    - Future Works
4. Summary

Examples of Scoring Function (SF)

Design principles

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



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]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r},\mathbf{w}_r\in\mathbb{R}^d$	$-\|(\mathbf{h}-\mathbf{w}_r^\top\mathbf{h}\mathbf{w}_r)+\mathbf{r}-(\mathbf{t}-\mathbf{w}_r^\top\mathbf{t}\mathbf{w}_r)\|_2^2$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1$
TransR [16]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^k, \mathbf{M}_r\in\mathbb{R}^{k\times d}$	$-\|\mathbf{M}_r\mathbf{h}+\mathbf{r}-\mathbf{M}_r\mathbf{t}\|_2^2$	$\|\mathbf{w}_r^\top\mathbf{r}\|/\|\mathbf{r}\|_2\le\epsilon, \|\mathbf{w}_r\|_2=1$ $\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$
TransD [50]	$\mathbf{h},\mathbf{w}_h\in\mathbb{R}^d$ $\mathbf{t},\mathbf{w}_t\in\mathbb{R}^d$	$\mathbf{r},\mathbf{w}_r\in\mathbb{R}^k$	$-\|(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}+\mathbf{r}-(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\|_2^2$	$\|\mathbf{M}_r\mathbf{h}\|_2\le 1, \|\mathbf{M}_r\mathbf{t}\|_2\le 1$ $\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$ $\|(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}\|_2\le 1$ $\|(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\|_2\le 1$
TranSparse [51]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^k, \mathbf{M}_r(\theta_r)\in\mathbb{R}^{k\times d}$ $\mathbf{M}_r^1(\theta_r^1), \mathbf{M}_r^2(\theta_r^2)\in\mathbb{R}^{k\times d}$	$-\|\mathbf{M}_r(\theta_r)\mathbf{h}+\mathbf{r}-\mathbf{M}_r(\theta_r)\mathbf{t}\|_{1/2}^2$ $-\|\mathbf{M}_r^1(\theta_r^1)\mathbf{h}+\mathbf{r}-\mathbf{M}_r^2(\theta_r^2)\mathbf{t}\|_{1/2}^2$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$ $\|\mathbf{M}_r(\theta_r)\mathbf{h}\|_2\le 1, \|\mathbf{M}_r(\theta_r)\mathbf{t}\|_2\le 1$ $\|\mathbf{M}_r^1(\theta_r^1)\mathbf{h}\|_2\le 1, \|\mathbf{M}_r^2(\theta_r^2)\mathbf{t}\|_2\le 1$
TransM [52]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$-\theta_r\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{1/2}$	$\|\mathbf{h}\|_2=1, \|\mathbf{t}\|_2=1$
ManifoldE [53]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$-(\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_2^2-\theta_r^2)^2$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$
TransF [54]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$(\mathbf{h}+\mathbf{r})^\top\mathbf{t}+(\mathbf{t}-\mathbf{r})^\top\mathbf{h}$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$
TransA [55]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d, \mathbf{M}_r\in\mathbb{R}^{d\times d}$	$-(\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|)^\top\mathbf{M}_r(\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|)$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$
KG2E [45]	$\mathbf{h}\sim\mathcal{N}(\boldsymbol{\mu}_h,\boldsymbol{\Sigma}_h)$ $\mathbf{t}\sim\mathcal{N}(\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)$	$\mathbf{r}\sim\mathcal{N}(\boldsymbol{\mu}_r,\boldsymbol{\Sigma}_r)$	$-\mathrm{tr}(\boldsymbol{\Sigma}_r^{-1}(\boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_t))-\boldsymbol{\mu}^\top\boldsymbol{\Sigma}_r^{-1}\boldsymbol{\mu}-\ln\frac{\det(\boldsymbol{\Sigma}_r)}{\det(\boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_t)}$ $-\boldsymbol{\mu}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}-\ln(\det(\boldsymbol{\Sigma}))$	$\|\mathbf{M}_r\|_F\le 1, [\mathbf{M}_r]_{ij}=[\mathbf{M}_r]_{ji}\ge 0$ $\|\boldsymbol{\mu}_h\|_2\le 1, \|\boldsymbol{\mu}_t\|_2\le 1, \|\boldsymbol{\mu}_r\|_2\le 1$ $c_{min}\mathbf{I}\le\boldsymbol{\Sigma}_h\le c_{max}\mathbf{I}$
TransG				
UM [56]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	—	$-\|\mathbf{h}-\mathbf{t}\|_2^2$	$\|\mathbf{h}\|_2=1, \|\mathbf{t}\|_2=1$
SE [57]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{M}_r^1,\mathbf{M}_r^2\in\mathbb{R}^{d\times d}$	$-\|\mathbf{M}_r^1\mathbf{h}-\mathbf{M}_r^2\mathbf{t}\|_1$	$\|\mathbf{h}\|_2=1, \|\mathbf{t}\|_2=1$

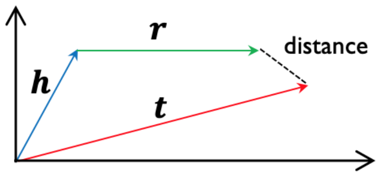
NTN [19]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r},\mathbf{b}_r\in\mathbb{R}^k, \underline{\mathbf{M}}_r\in\mathbb{R}^{d\times d\times k}$ $\mathbf{M}_r^1,\mathbf{M}_r^2\in\mathbb{R}^{k\times d}$	$\mathbf{r}^\top\tanh(\mathbf{h}^\top\underline{\mathbf{M}}_r\mathbf{t}+\mathbf{M}_r^1\mathbf{h}+\mathbf{M}_r^2\mathbf{t}+\mathbf{b}_r)$	$\|\mathbf{b}_r\|_2\le 1, \|\underline{\mathbf{M}}_r^{[:,:,i]}\|_F\le 1$ $\|\mathbf{M}_r^1\|_F\le 1, \|\mathbf{M}_r^2\|_F\le 1$
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]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$f_r(h,t)=\mathbf{t}^\top\mathbf{z}^{(L)}$ $\mathbf{z}^{(\ell)}=\mathrm{ReLU}(\mathbf{a}^{(\ell)}), \ \mathbf{a}^{(\ell)}=\mathbf{M}^{(\ell)}\mathbf{z}^{(\ell-1)}+\mathbf{b}^{(\ell)}$ $\mathbf{z}^{(0)}=[\mathbf{h};\mathbf{r}]$	—

Too many existing ones, hard to design

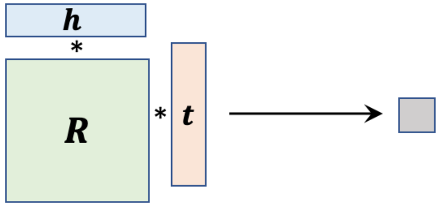
Wang et.al. Knowledge graph embedding: A survey of approaches and applications. TKDE 2017

Examples:

1. Translation Distance Models (TDMs)
   - TransE, TransH, RotatE, etc
   - less expressive

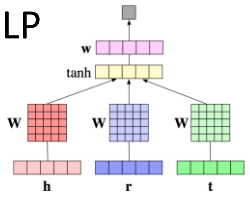


2. BiLinear Models (BLMs)
   - DistMult, ComplEx, Analogy, SimplE, etc
   - state-of-the-art and fully expressive

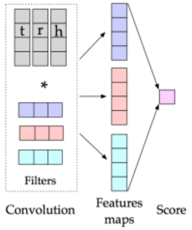


3. Neural Network Models (NNMs)
   - MLP, ConvE, RSN, etc
   - complex and difficult to train

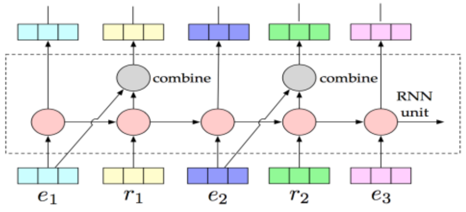
MLP



ConvE



RSN



Contribution – Search to capture semantics

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 (i.e., prior on semantics) is important
3. Designing novel and universal SFs becomes harder

Our solutions:

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

AutoSF: Searching Scoring Functions for Knowledge Graph Embedding. ICDE 2020

Revisit Bilinear SFs

DistMult: Embedding entities and relations for learning and inference in knowledge bases. ICLR 2015
ComplEx: Knowledge graph completion via complex tensor factorization. JMLR 2017
Analogy: Analogical inference for multirelational embeddings. ICML 2018
SimplE embedding for link prediction in knowledge graphs. NeurIPS 2018

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

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

symmetric	√
anti-symmetric	√
asymmetric	√
inverse	√

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

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

symmetric	√
anti-symmetric	√
asymmetric	√
inverse	√

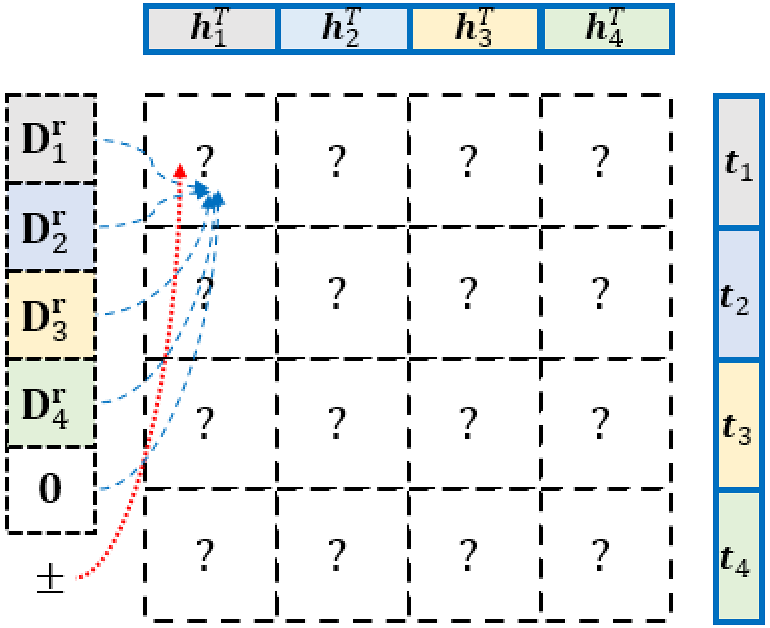
AutoSF: Search to regularize bilinear SFs

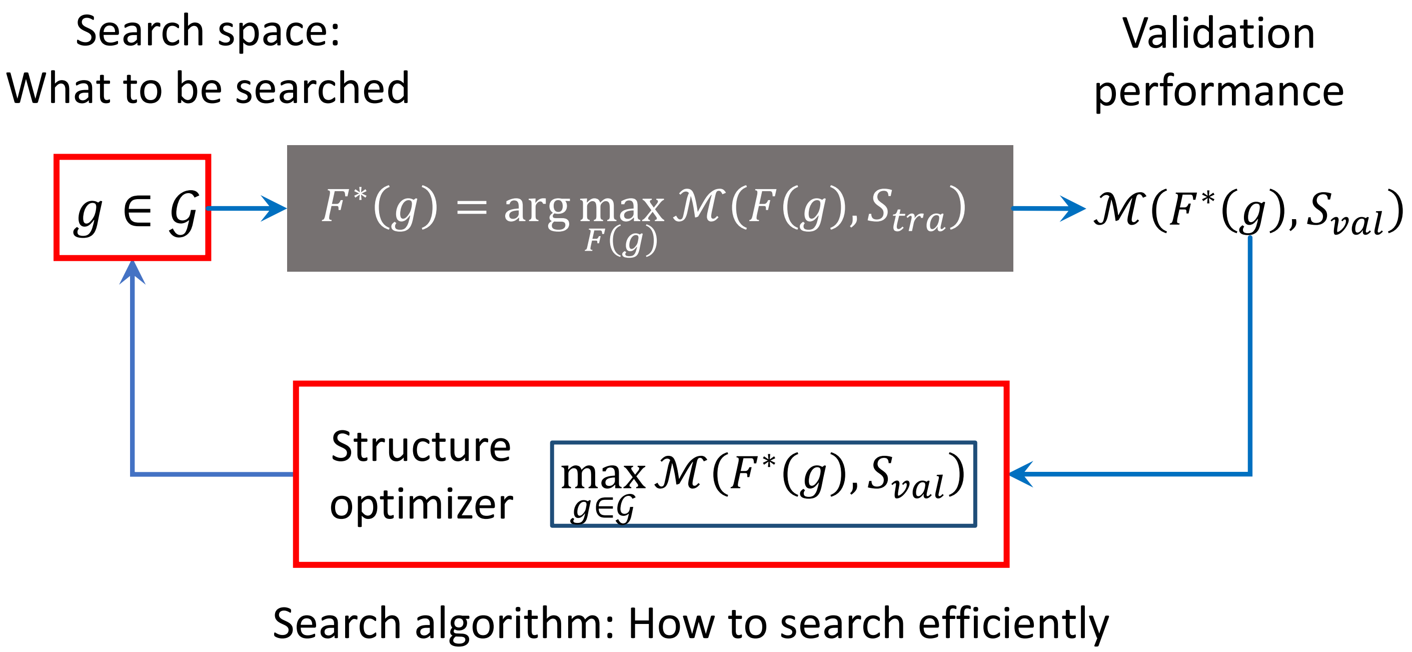
Definition 3 (S2R Problem). *Let $F(\boldsymbol{P}; g)$ be a KG embedding model (with indexed embeddings $\boldsymbol{P} = \{\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}\}$ and architecture g), $M(F, \mathcal{S})$ measures the performance of a KG embedding model F on a set of triplets $\mathcal{S}$ (the higher the better). The problem of S2R is formulated as:*

$$g^* \in \arg\max_{g \in \mathcal{G}} M\left(F(\boldsymbol{P}^*; g), \mathcal{S}_{val}\right) \quad (4)$$

$$\text{s.t.} \boldsymbol{P}^* = \arg\max_{\boldsymbol{P}} M(F(\boldsymbol{P}; g), \mathcal{S}_{tra}), \quad (5)$$

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





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

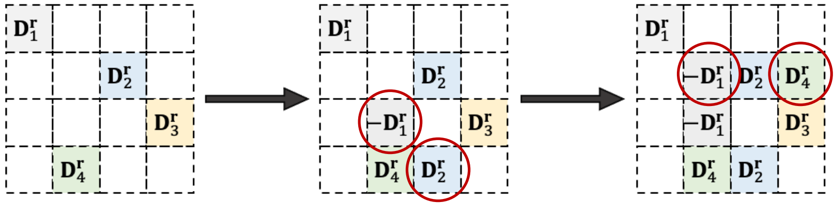
The location of a block matrix $\mathbf{D}_i^r$ represents a multiplicative term.

AutoSF – Search algorithm

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?

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



Not all scoring functions / structures need to be trained.

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

➢ Filter: remove bad and equivalent SFs.

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

Select better SFs based on matrix structure to train and evaluate.

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.

Experiments – Effectiveness

type	model	WN18			FB15k			WN18RR			FB15k237			YAGO3-10		
		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**

Measurements

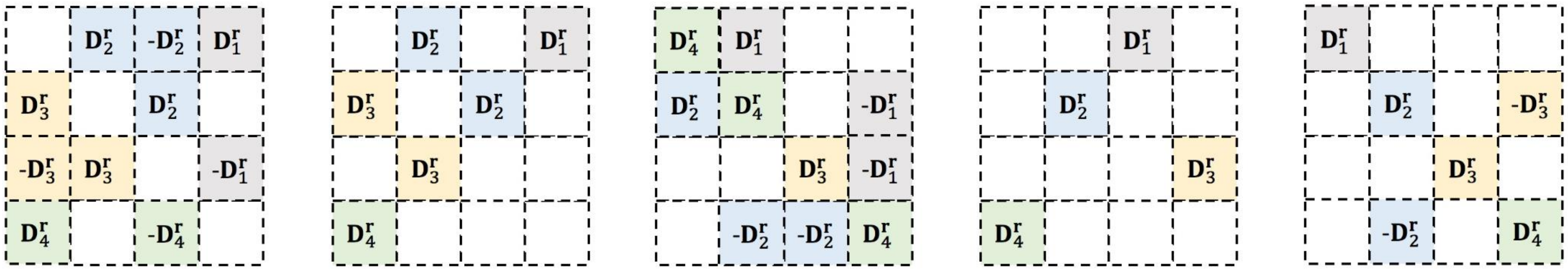
- Given a triplet (h, r, t);
- Compute the score of $(h', r, t), \forall h' \in \mathcal{E}$;
- Get the rank of h among all h'

Metrics

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

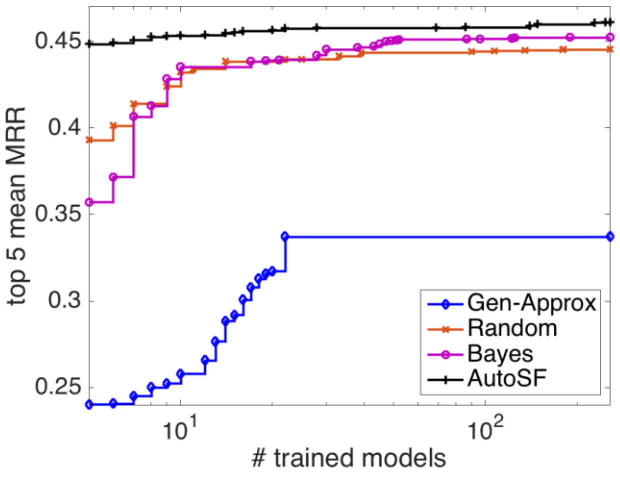
- BLMs are better than the other types and rule-based models
- There is no absolute winner among the BLMs
- Compared with human-designed ones, the SFs searched by AutoSF always lead the performance

Experiments – Efficiency

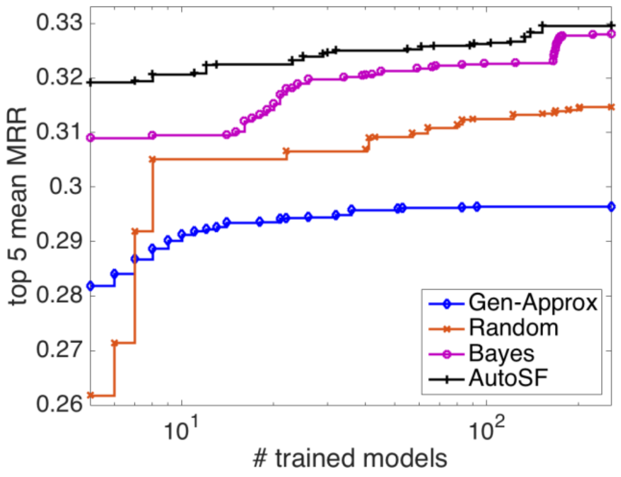


(a) WN18. (b) FB15k. (c) WN18RR. (d) FB15k237. (e) YAGO3-10.

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



WN18-RR



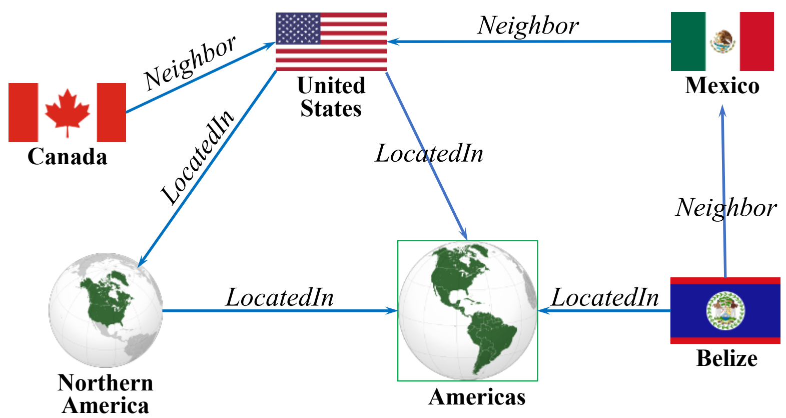
FB15k237

- 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

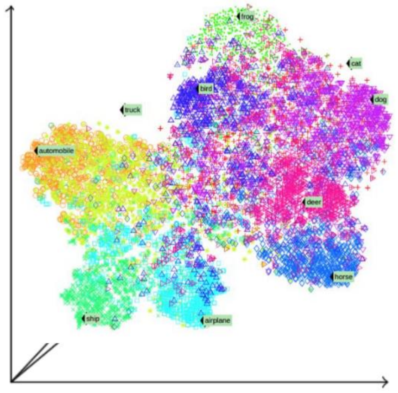
1. What is Knowledge Graph (KG)?
2. What is Automated Machine Learning (AutoML)?
3. Attacking Core Issues in KG by AutoML
   - Overview of Ideas
   - ICDE 2020: Search to Capture Semantics
   - NeurIPS 2020: Search to Exploit Graph Topology
   - Future Works
4. Summary

Relational Path in KG



vectorize

entities & relations



Triplets: (s, r, o);

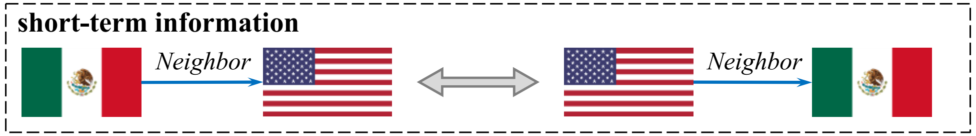
Relational path [1,2]:

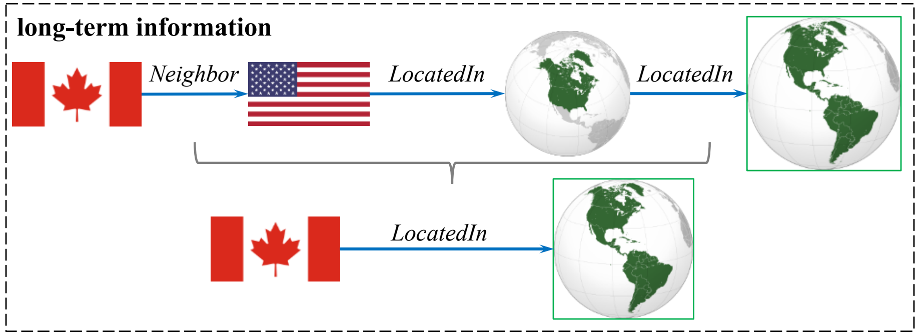
$$s_1 \xrightarrow{r_1} s_2 \xrightarrow{r_2} s_3 \cdots s_L \xrightarrow{r_L} s_{L+1}$$

Short-term information inside triplets.

Composition of relations.

Long-term information across triplets.





[1]. Guu et.al. Traversing knowledge graphs in vector space. EMNLP, 2015

[2]. Sadeghian et.al. DRUM: End-to-end differentiable rule mining on knowledge graphs. NeurIPS 2019

Contribution – Search to exploit topology

1. The relational path contains several mixed information.
2. Link prediction task emphasizes on the short-term semantic information, while entity alignment task requires to model the long-term information.
3. How to properly encode such prior knowledge into the model design?

Our solutions:

- Search to adaptively learn the mixed information in relational path.
- A novel hybrid-search algorithm for efficient search.

Interstellar: Searching Recurrent Architecture for Knowledge Graph Embedding. NeurIPS 2020

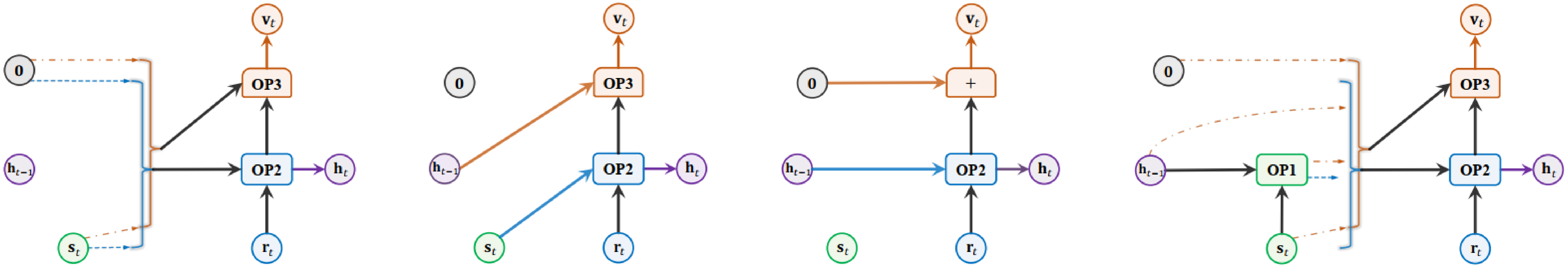
Recurrent Structure – Case study

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

harder longer

Table 3: Performance on Countries dataset.

	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**
Interstellar	**1.000±0.000**	**1.000±0.000**	**0.968± 0.007**

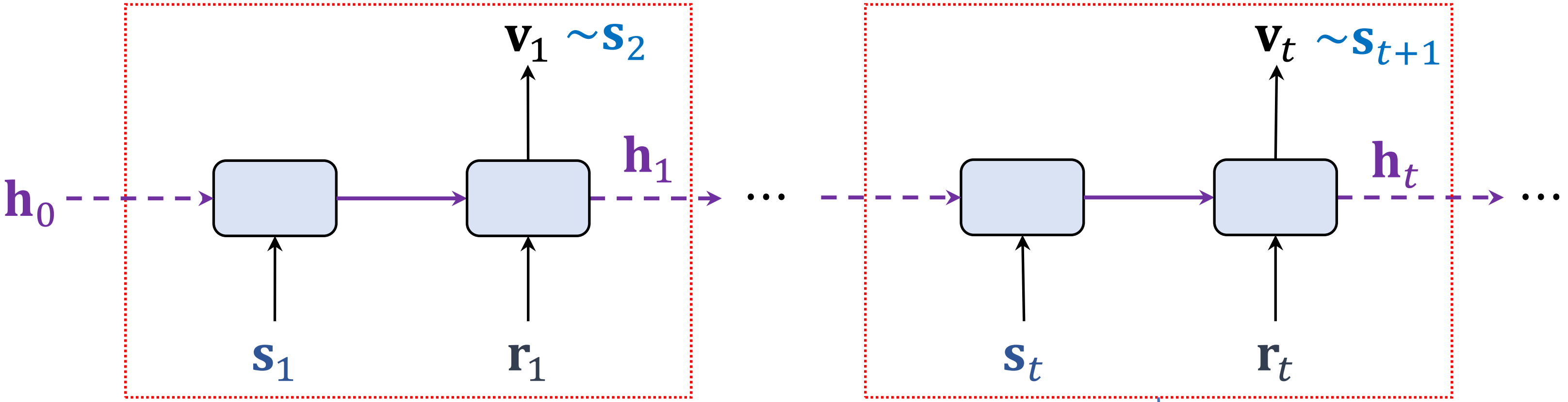


P1: single triplet P2: two triples P3: relation composition P4: fully recurrent

Model design should be data-specific. Search to leverage proper prior knowledge.

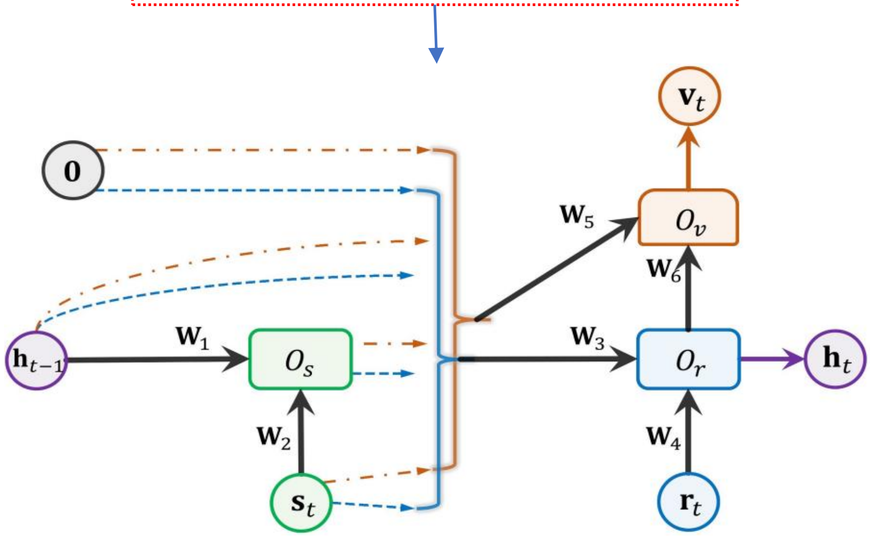
Interstellar: Searching recurrent strcture

Recurrently process the path by $[\mathbf{v}_t, \mathbf{h}_t] = f(\mathbf{s}_t, \mathbf{r}_t, \mathbf{h}_{t-1}), \ \ \forall t = 1 \ldots L$



Searching !

macro-level	connections	$\mathbf{h}_{t-1}, O_s, \mathbf{0}, \mathbf{s}_t$
$\hat{\boldsymbol{\alpha}} \in \hat{\mathcal{A}}$	combinators	$+, \odot, \otimes$, gated
micro-level	activation	identity, tanh, sigmoid
$\breve{\boldsymbol{\alpha}} \in \breve{\mathcal{A}}$	weight matrix	$\{\mathbf{W}_i\}_{i=1}^{6}, \mathbf{I}$



Hybrid search algorithm

Search appropriate $\alpha \in \mathcal{A}$ that maximize the validation performance

$$\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \mathcal{M}\left(f(\boldsymbol{F}^*;\boldsymbol{\alpha}), \mathcal{G}_{\text{val}}\right), \quad \text{s.t.} \quad \boldsymbol{F}^* = \arg\min_{\boldsymbol{F}} \mathcal{L}\left(f(\boldsymbol{F};\boldsymbol{\alpha}), \mathcal{G}_{\text{tra}}\right)$$

Stand-alone approach:

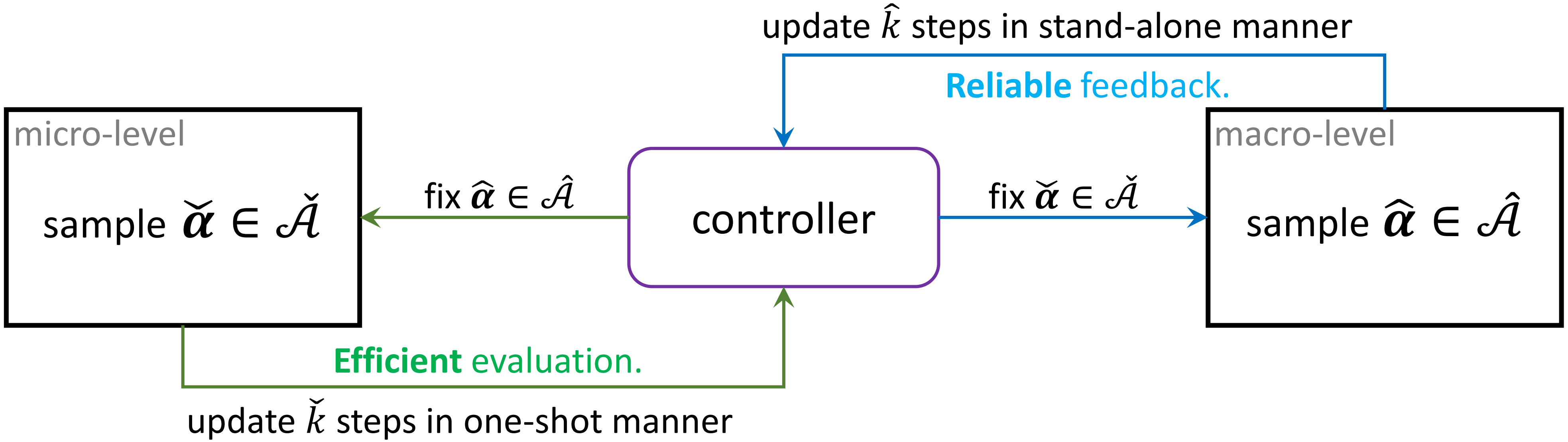
- $\mathcal{M}$ is accurate;
- $\boldsymbol{F}^*$ needs high cost.

[Zoph and Le 2017]

One-shot approach:

- $\boldsymbol{F}^*$ is shared and efficient;
- $\mathcal{M}$ is not always reliable.

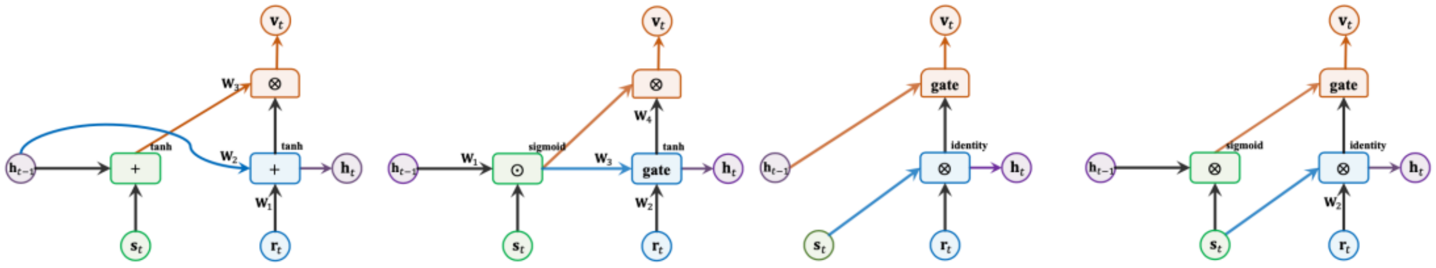
[Pham et al. 2018, Liu et al. 2019]



Experiments – Effectiveness

Entity alignment task

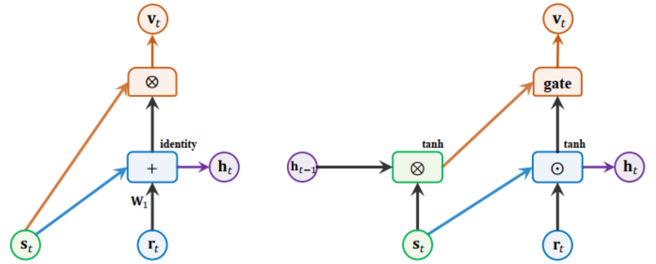
	models	DBP-WD H@1	DBP-WD H@10	DBP-WD MRR	DBP-YG H@1	DBP-YG H@10	DBP-YG MRR	EN-FR H@1	EN-FR H@10	EN-FR MRR	EN-DE H@1	EN-DE H@10	EN-DE MRR
triplet	TransE	18.5	42.1	0.27	9.2	24.8	0.15	16.2	39.0	0.24	20.7	44.7	0.29
	TransD*	27.7	57.2	0.37	17.3	41.6	0.26	21.1	47.9	0.30	24.4	50.0	0.33
	BootEA*	32.3	63.1	0.42	31.3	62.5	0.42	31.3	62.9	0.42	44.2	70.1	0.53
GCN	GCN-Align	17.7	37.8	0.25	19.3	41.5	0.27	15.5	34.5	0.22	25.3	46.4	0.22
	VR-GCN	19.4	55.5	0.32	20.9	55.7	0.32	16.0	50.8	0.27	24.4	61.2	0.36
	R-GCN	8.6	31.4	0.16	13.3	42.4	0.23	7.3	31.2	0.15	18.4	44.8	0.27
path	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
	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
	Chains	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
	SRAP	**40.7**	**71.2**	**0.51**	**40.2**	**72.0**	**0.51**	**35.5**	**67.9**	**0.46**	**50.1**	**75.6**	**0.59**



(a) DBP-WD (b) DBP-YG (c) EN-FR (d) EN-DE

Link prediction task

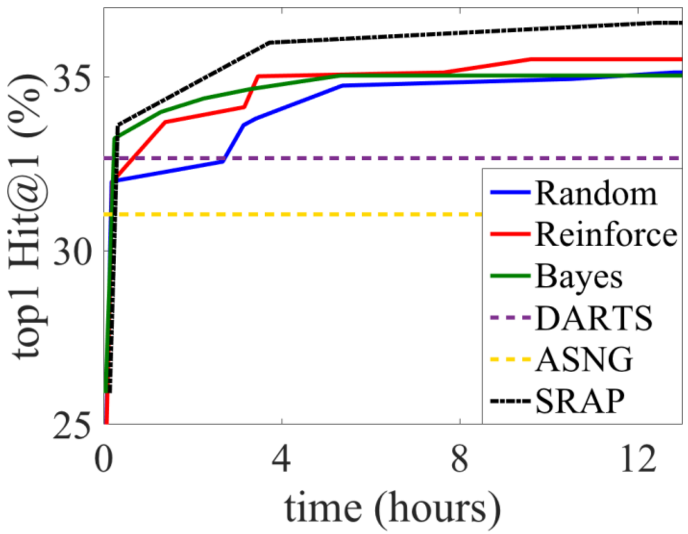
models	WN18-RR H@1	WN18-RR H@10	WN18-RR MRR	FB15k-237 H@1	FB15k-237 H@10	FB15k-237 MRR
TransE	12.5	44.5	0.18	17.3	37.9	0.24
ComplEx	41.4	49.0	0.44	22.7	49.5	0.31
RotatE	43.6	54.2	0.47	**23.3**	50.4	**0.32**
R-GCN	-	-	-	15.1	41.7	0.24
PTransE	27.2	46.4	0.34	20.3	45.1	0.29
RSN	38.0	44.8	0.40	19.2	41.8	0.27
SRAP	**44.0**	**54.8**	**0.48**	**23.3**	**50.8**	**0.32**



WN18RR FB15k-237

Experiments – Efficiency

entity alignment



link prediction

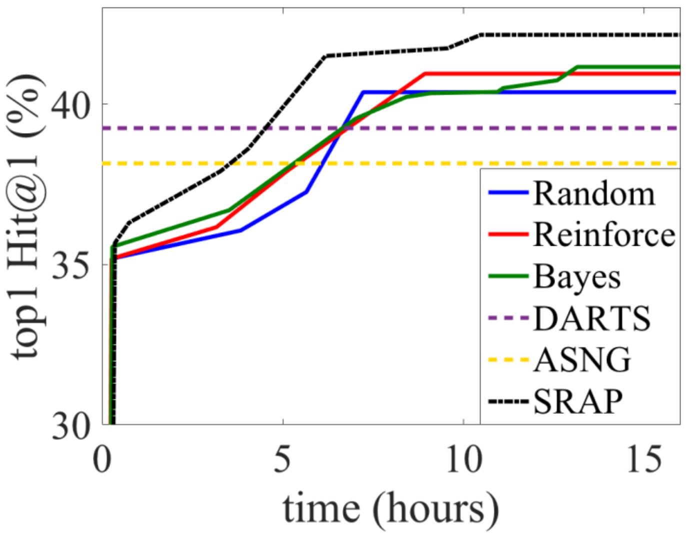


Table 6: Comparison of searching and fine-tuning time (in hours) in Algorithm 1.

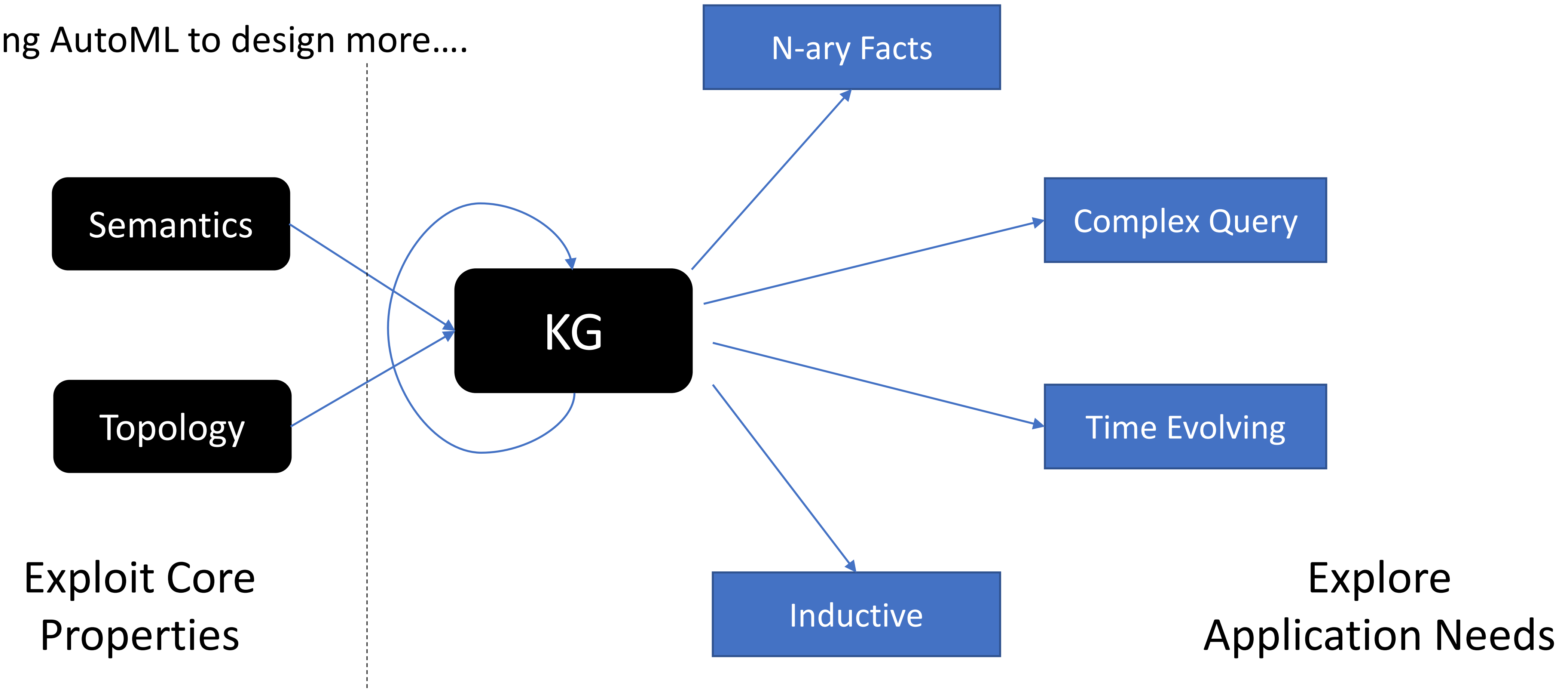
procedure		entity alignment		link prediction	
		Normal	Dense	WN18-RR	FB15k-237
search	macro-level (line 2-3)	9.9±1.5	14.9±0.3	11.7±1.9	23.2±3.4
	micro-level (line 4-5)	4.2±0.2	7.5±0.6	6.3± 0.9	5.6±0.4
fine-tune (line 7)		11.6±1.6	16.2±2.1	44.3±2.3	67.6±4.5

Outline

1. What is Knowledge Graph (KG)?
2. What is Automated Machine Learning (AutoML)?
3. Attacking Core Issues in KG by AutoML
   - Overview of Ideas
   - ICDE 2020: Search to Capture Semantics
   - NeurIPS 2020: Search to Exploit Graph Topology
   - Future Works
4. Summary

Future works

Using AutoML to design more....



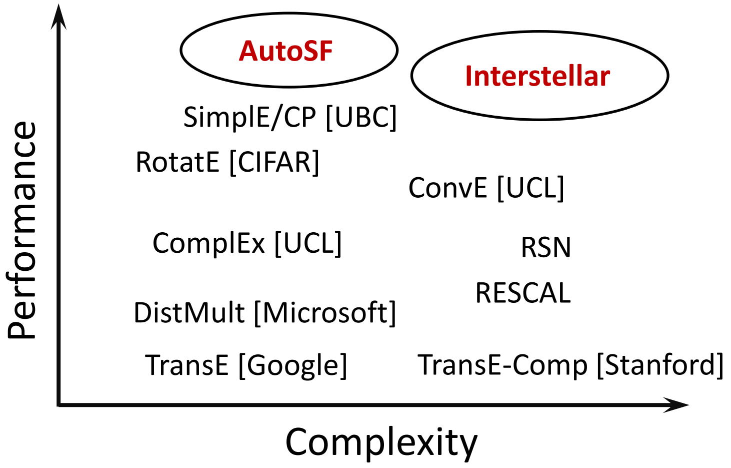
Outline

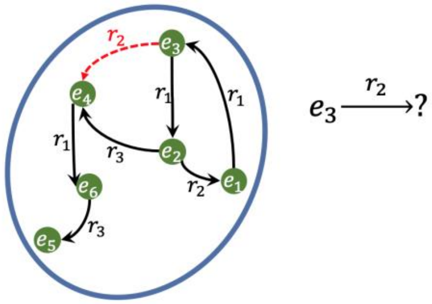
1. What is Knowledge Graph (KG)?
2. What is Automated Machine Learning (AutoML)?
3. Attacking Core Issues in KG by AutoML
4. Summary

Summary

Design data-specific KG learning methods by AutoML

- Better explore semantics and topology
- Adapt to different application needs





Code: https://github.com/AutoML-4Paradigm

Related Publications

1. Efficient Relation-aware Scoring Function Search for Knowledge Graph Embedding. ICDE 2021
2. Efficient, Simple and Automated Negative Sampling for Knowledge Graph Embedding. VLDBJ 2020.
3. Interstellar: Searching Recurrent Architecture for Knowledge Graph Embedding. NeurIPS 2020
4. Generalizing Tensor Decomposition for N-ary Relational Knowledge Bases. WWW 2020
5. AutoSF: Searching Scoring Functions for Knowledge Graph Embedding. ICDE 2020

Joint works with: Yongqi Zhang (*4Paradigm*), Yu Liu (*Tshinghua*), Shimin Di (*HKUST*)

Thanks!

恳请各位批评&指正!