ECAI DIGITAL - 2020

kge-tutorial-ecai2020.github.io

Knowledge Graph Embeddings: From Theory to Practice

Luca Costabello
Accenture Labs
@lukostaz

Sumit Pai
Accenture Labs
@sumitppai

Nicholas McCarthy
Accenture Labs
@nickpmcc

Adrianna Janik
Accenture Labs
@adri_janik

Outline

Theoretical Overview 1h 30m

- Introduction
- Anatomy of a Knowledge Graph Embedding Model
- Evaluation Protocol and Metrics
- Advanced KGE Topics
- Open Research Questions

Applications 15 m

Software Ecosystem 15 m

Hands-on Session 1h 15m

Outline

Theoretical Overview — 1h 30m

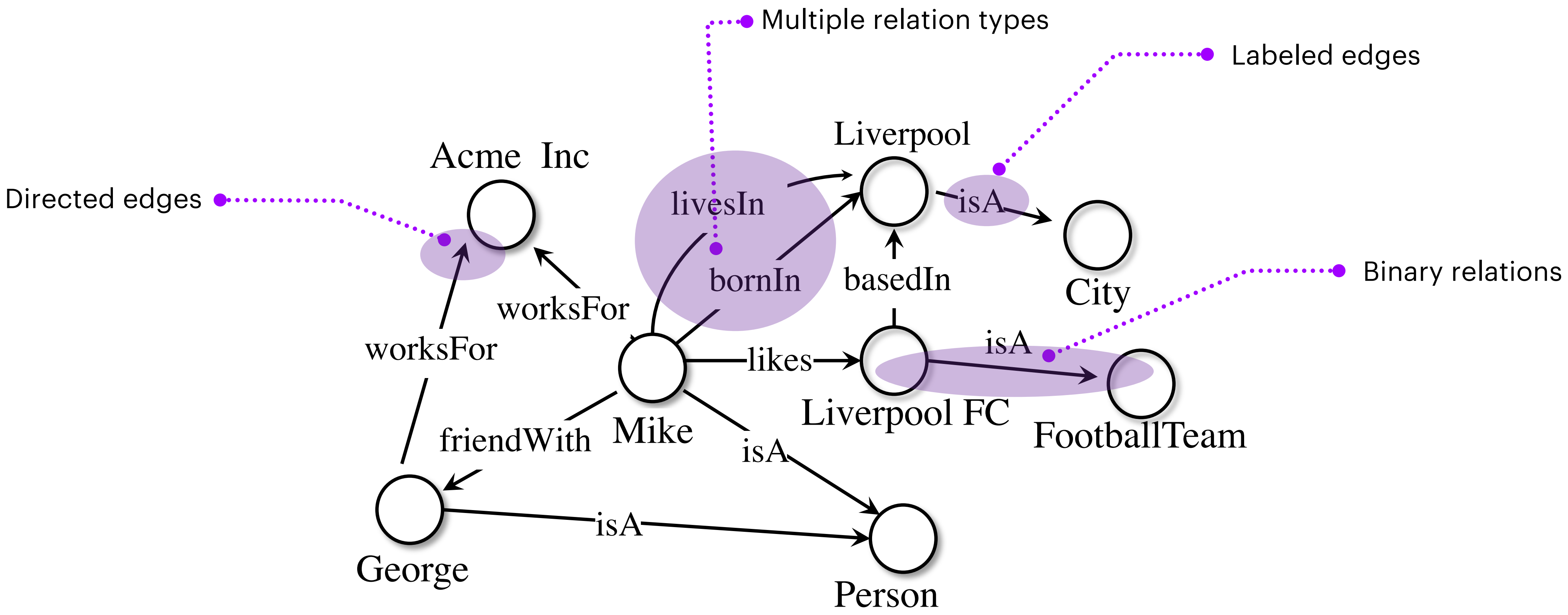
- **Introduction**
- Anatomy of a Knowledge Graph Embedding Model
- Evaluation Protocol and Metrics
- Advanced KGE Topics
- Open Research Questions

Applications — 15 m

Software Ecosystem — 15 m

Hands-on Sessions — 1h 15m

Knowledge Graph

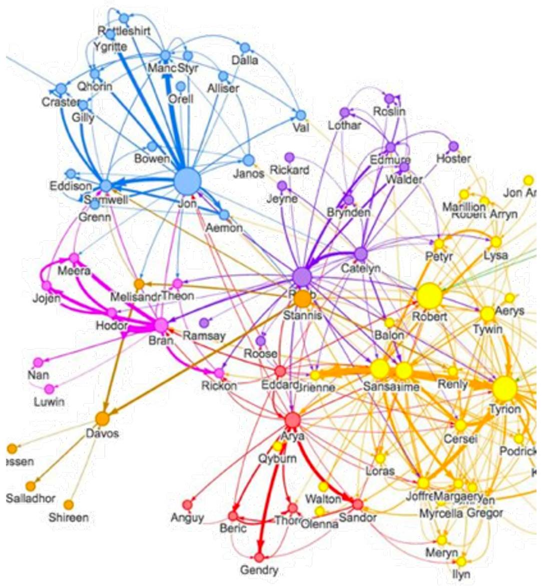


$$\mathcal{G} = \{(s, p, o)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$$

$\mathcal{E}$: set of entities of $\mathcal{G}$ $\mathcal{R}$: set of relations of $\mathcal{G}$

In-depth overview of Knowledge Graphs in [Hogan et al. 2020]

SOCIAL NETWORKS

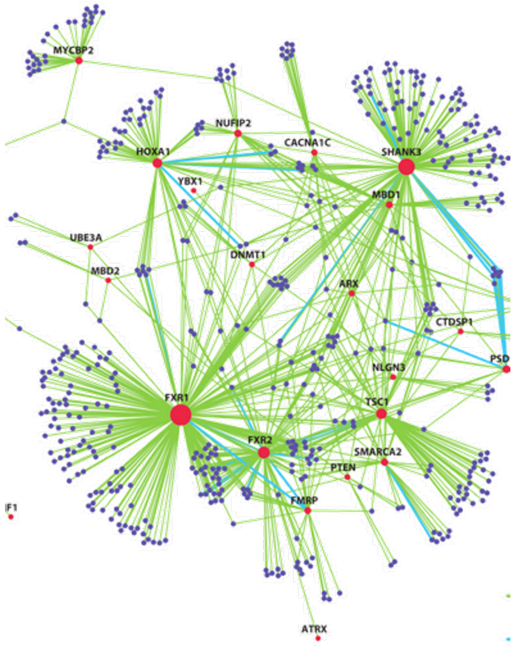


[neo4j.com]

COLLABORATIVE WEB-BASED KNOWLEDGE BASES

[lod-cloud.net]

PROTEIN-PROTEIN INTERACTION NETWORKS



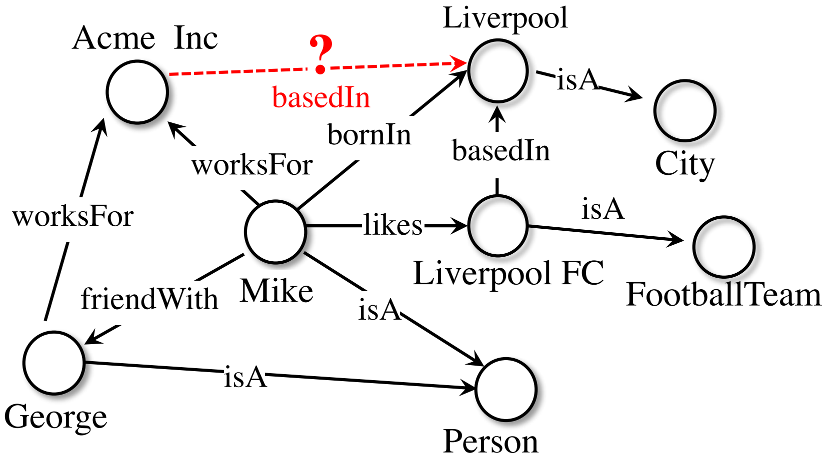
[ebi.ac.uk]

Knowledge Graph	Statements	Entities
yago select knowledge	120 M	10 M
WIKIDATA	610 M	51 M
DBpedia	1.3 B	6 M
GDELT	3.5 B	364 M

Knowledge Graphs & The Open World Assumption

- **Closed World Assumption (CWA)**: absence of a fact means it is necessarily false.
- **Open World Assumption (OWA)**: absence of a fact does not imply fact is false. We simply do not know.

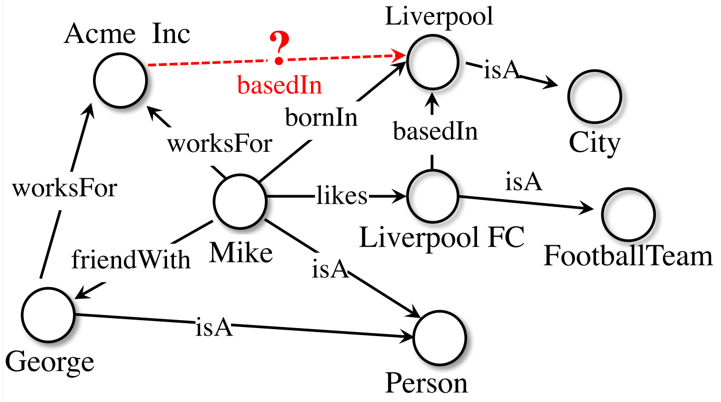
Knowledgee Graphs
adopt this assumption



Machine Learning on Knowledge Graphs/ Statistical Relational Learning

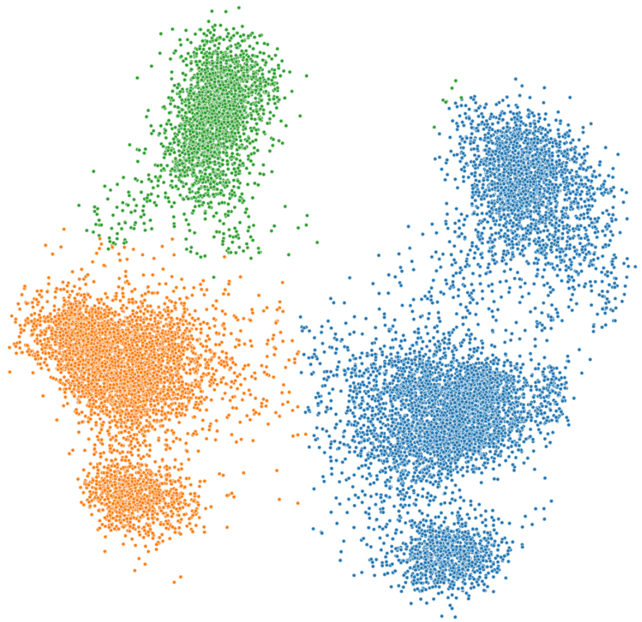
LINK PREDICTION / TRIPLE CLASSIFICATION

- Knowledge graph completion
- Content recommendation
- Question answering



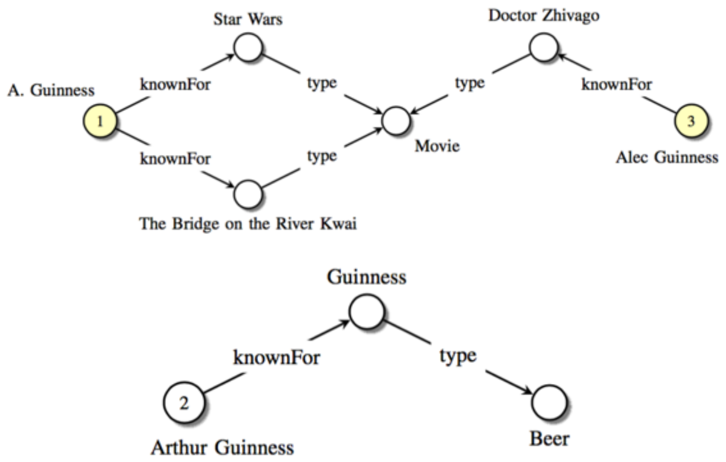
COLLECTIVE NODE CLASSIFICATION / LINK-BASED CLUSTERING

- Customer segmentation



ENTITY MATCHING

- Duplicate detection
- Inventory items deduplication

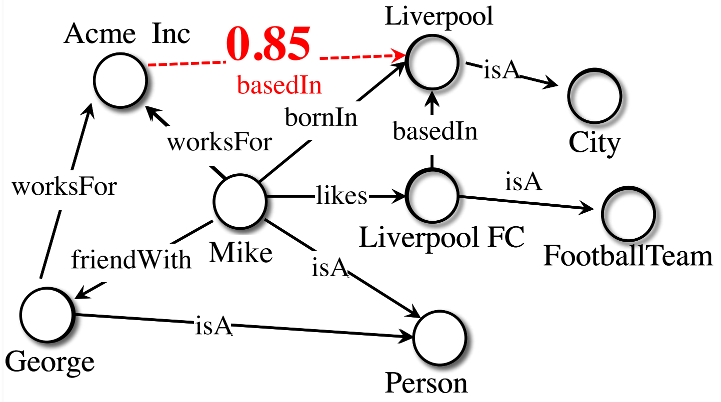


Pic from [Nickel et al. 2016a]

Machine Learning on Knowledge Graphs/ Statistical Relational Learning

LINK PREDICTION / TRIPLE CLASSIFICATION

- Knowledge graph completion
- Content recommendation
- Question answering



Assigning a score proportional to the likelihood that an unseen triple is true.

Link Prediction

- Learning to rank problem
- Information retrieval metrics
- No ground truth negatives in test set required

Triple Classification

- Binary classification task
- Binary classification metrics
- Test set requires positives and ground truth negatives

Traditional Statistical Relational Learning

- **Logic Programming**: predict new links from facts and extracted rules.
- **Inductive Logic Programming (ILP)**: predict new links from rules extracted from correlated facts.
- **Rule Mining**: e.g. AMIE+: extracts Horn clauses based on their support in the KG.

[Gallaraga et al. 2015]

- **Graphical Models**:
  - Conditional Random Fields (CRFs)
  - Probabilistic Relational Models
  - Relational Markov Networks
  - Relational Dependency Networks

[Getoor & Taskar 2007]

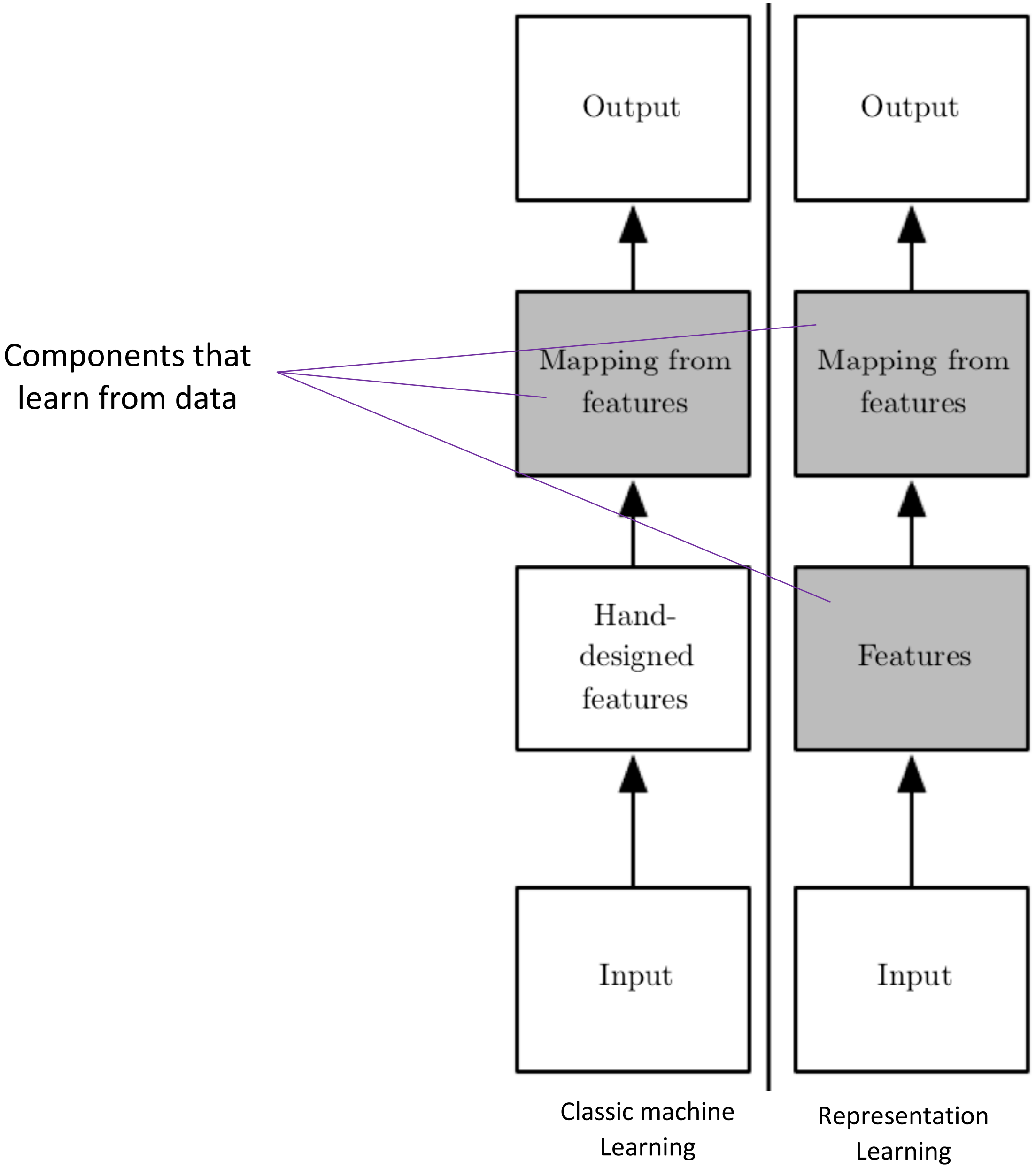
Limitations

- Limited Scalability to KG size
- Limited modeling power
- Non-differentiable approaches

Introducing (Graph) Representation Learning

- Manual feature engineering on graphs is hard and time-consuming…

- … what if instead we *learn* representations of nodes and edges?



Pic from [Goodfellow et al. 2016]

Can we re-use traditional deep learning tools?

- CNNs are designed for grids (e.g. images)
- RNNs/word2vec for sequences (e.g. text)

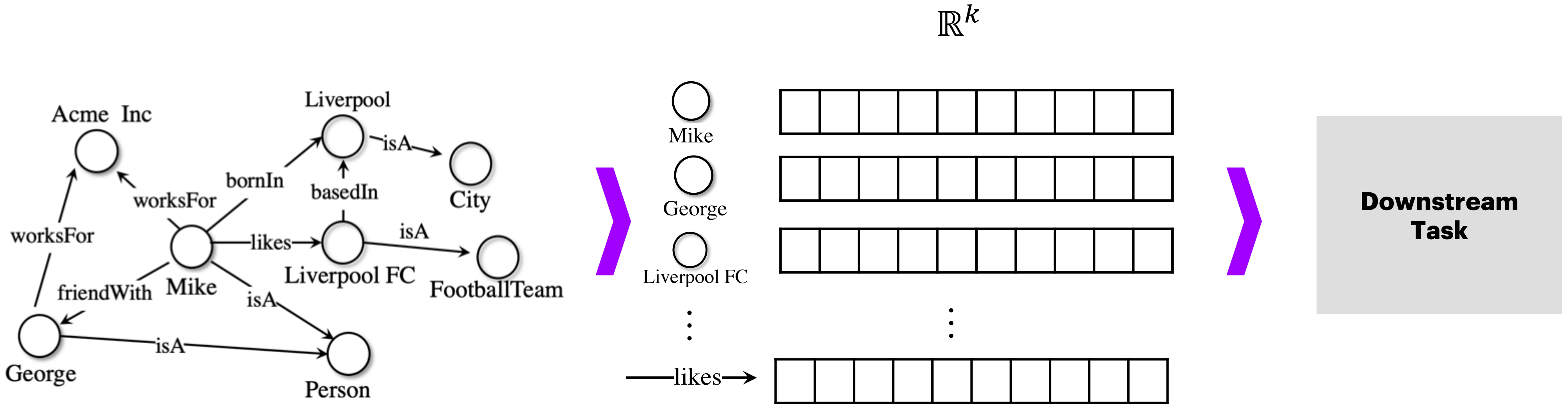
But graphs are more complex:

- No spatial locality
- No fixed-node ordering (graph isomorphism problem)
- Multimodal (concepts, text, numbers, timestamps, etc.)

We need ad-hoc models!

From AAAI-19 tutorial on Graph Representation Learning [Hamilton & Tang 2019]

Graph Representation Learning
Learning representations of nodes and edges



Node Representation/Graph Feature based Methods
PRA, LINE, DeepWalk, node2vec

Graph Neural Networks (GNNs)
GCNs, Graph Attention Networks

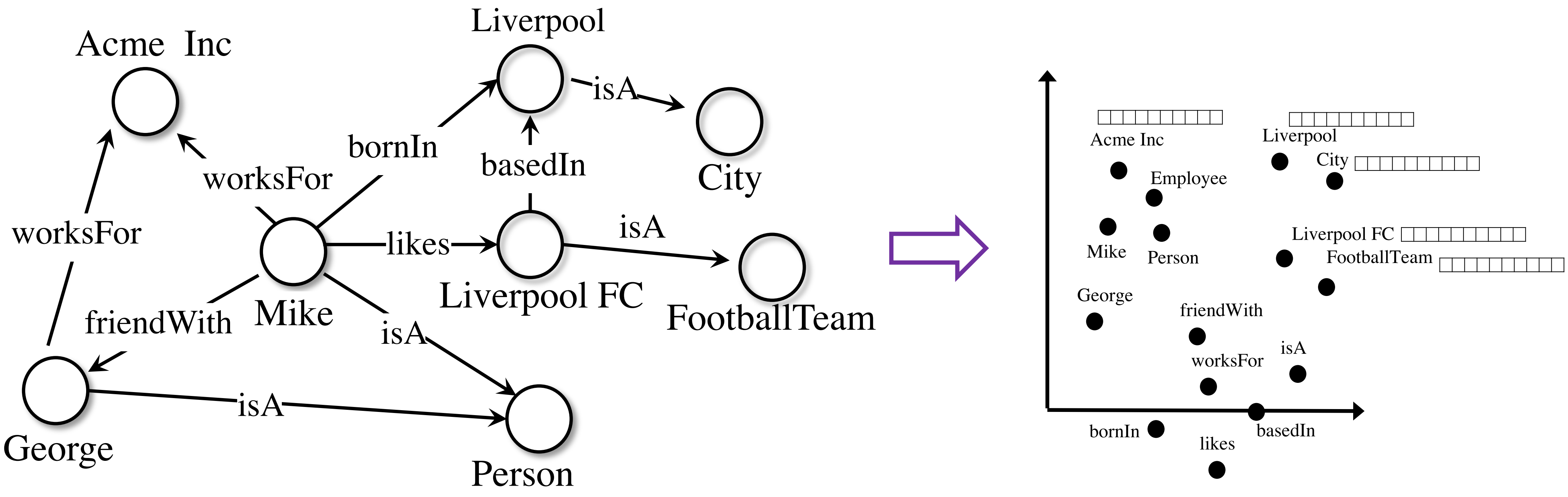
Knowledge Graph Embeddings (KGE)
TransE, DistMult, ComplEx, ConvE

Scope of this tutorial

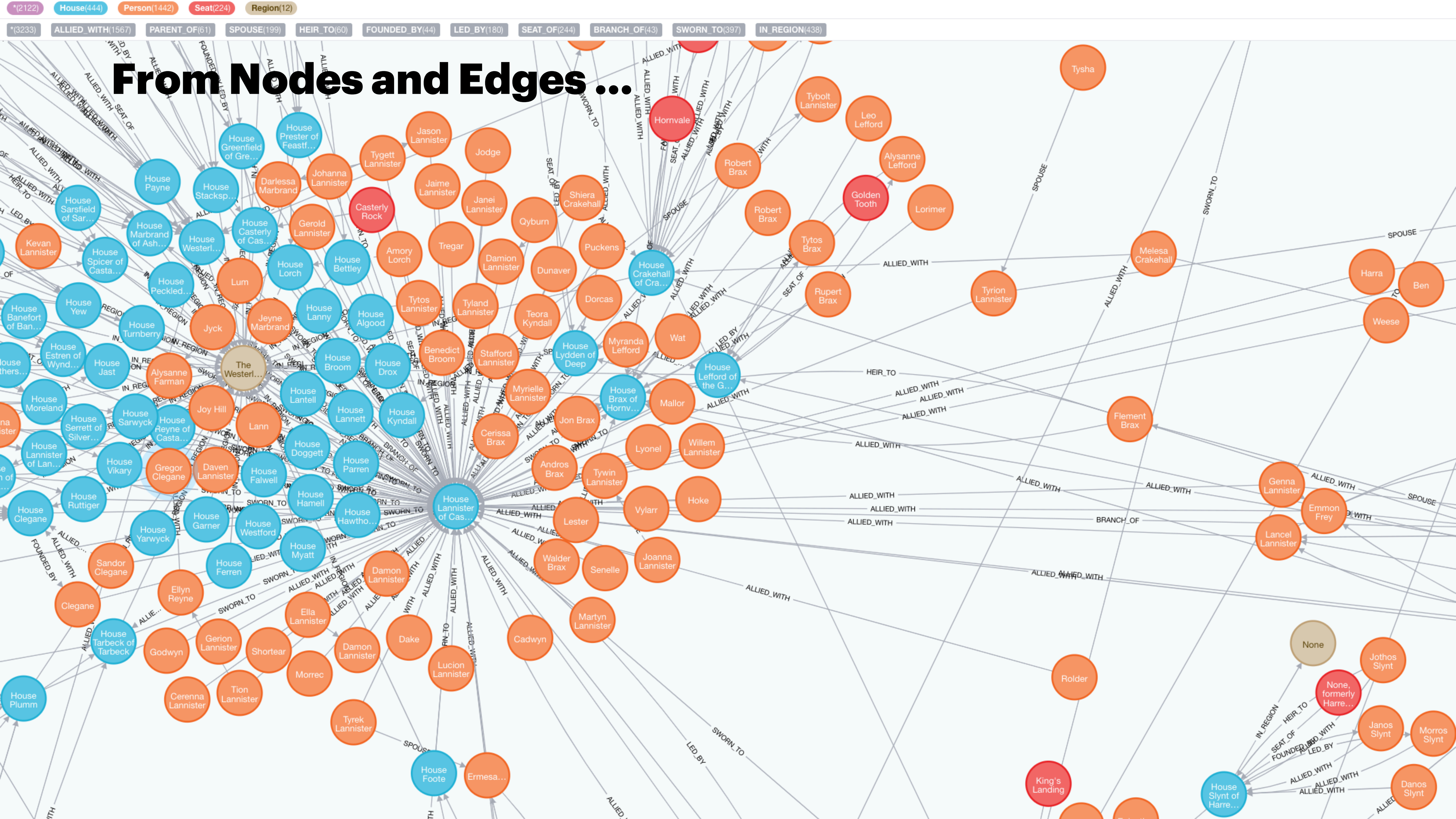
For a complete overview of graph feature-based models and GNNs:
[Hamilton & Sun 2019]
[Hamilton 2020]

Knowledge Graph Embeddings (KGE)

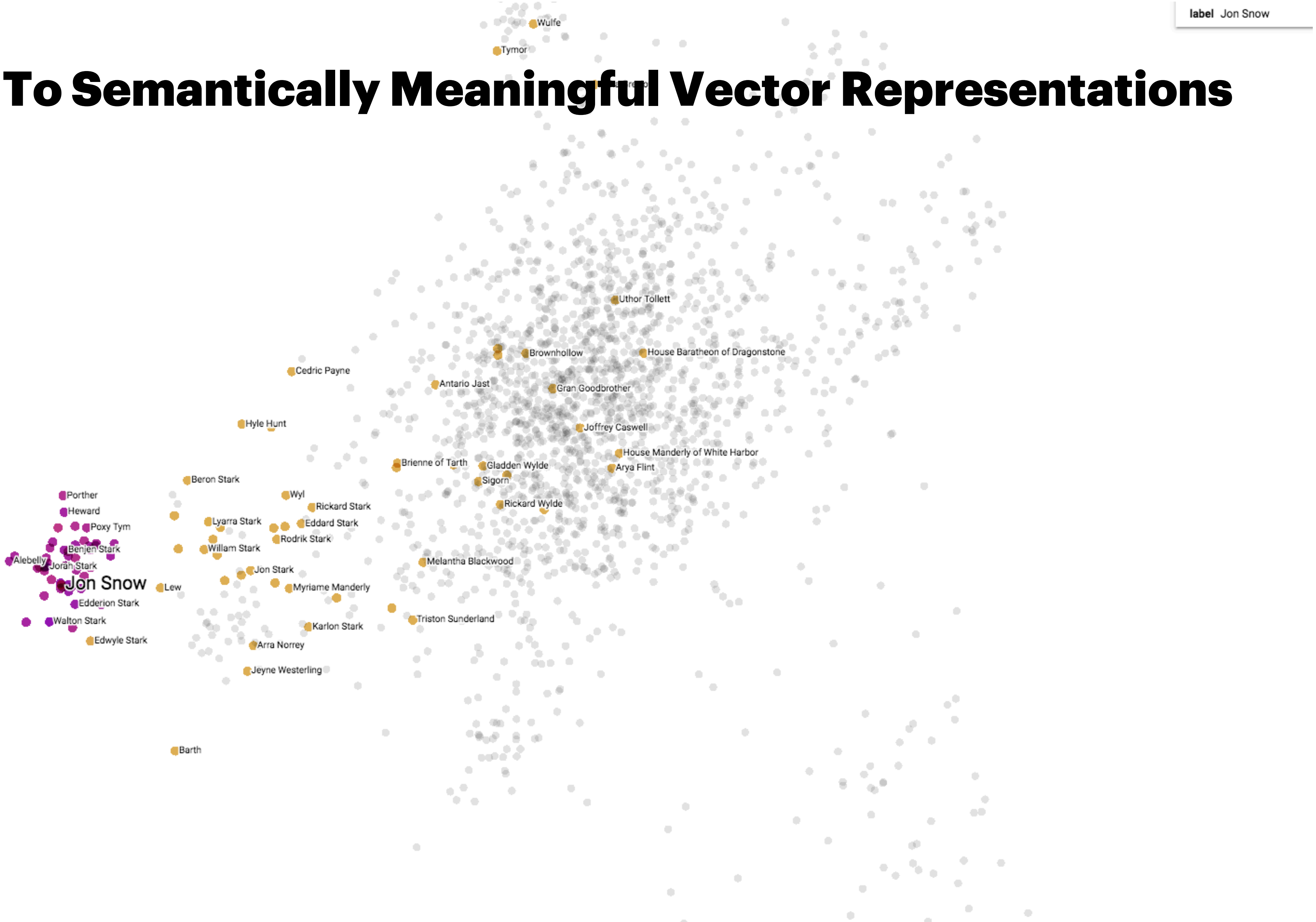
Automatic, supervised learning of **embeddings**, i.e. projections of entities and relations into a continuous low-dimensional space.



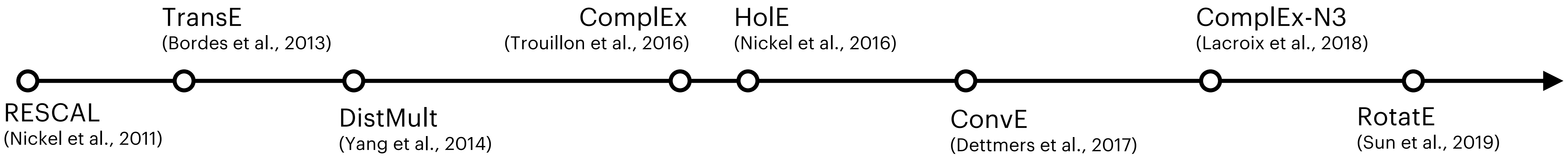
From Nodes and Edges ...



… To Semantically Meaningful Vector Representations



(Some) KGE models in recent published literature:



KGE Design Rationale: Capture KG Patterns

Symmetry	`<Alice marriedTo Bob>`
Asymmetry	`<Alice childOf Jack>`
Inversion	`<Alice childOf Jack>` `<Jack fatherOf Alice>`

Composition	`<Alice childOf Jack>`
	`<Jack siblingOf Mary>`
	`<Alice nieceOf Mary>`

But also:
- Hierarchies
- Type constraints
- Transitivity
- Homophily
- Long-range dependencies

Model	Symmetry	Antisymmetry	Inversion	Composition
SE	✗	✗	✗	✗
TransE	✗	✓	✓	✓
TransX	✓	✓	✗	✗
DistMult	✓	✗	✗	✗
ComplEx	✓	✓	✓	✗
RotatE	✓	✓	✓	✓

[Sun et al. 2019]

Outline

Theoretical Overview 1h 30m

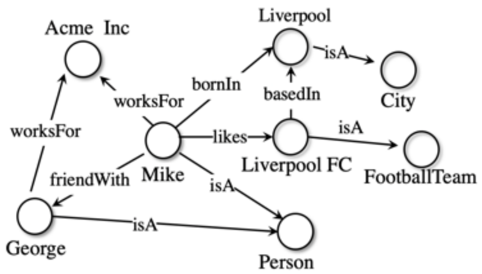
- Introduction
- **Anatomy of a Knowledge Graph Embedding Model**
- Evaluation Protocol and Metrics
- Advanced KGE Topics
- Open Research Questions

Applications 15 m

Software Ecosystem 15 m

Hands-on Sessions 1h 15m

At a Glance



Optimizer

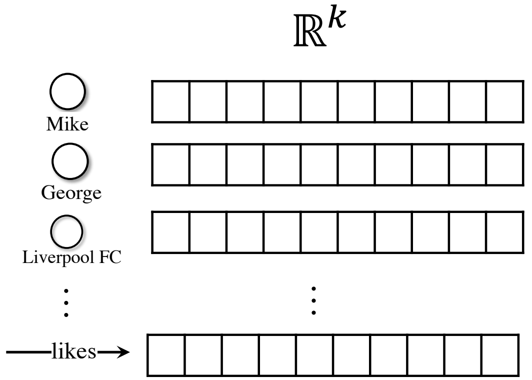
Lookup Layer

Scoring Layer $\boldsymbol{f(s,p,o)} \in \mathbb{R}$

Loss Functions $\boldsymbol{\mathcal{L}}$

Negatives Generation

Training



Downstream Tasks
(Link Prediction)

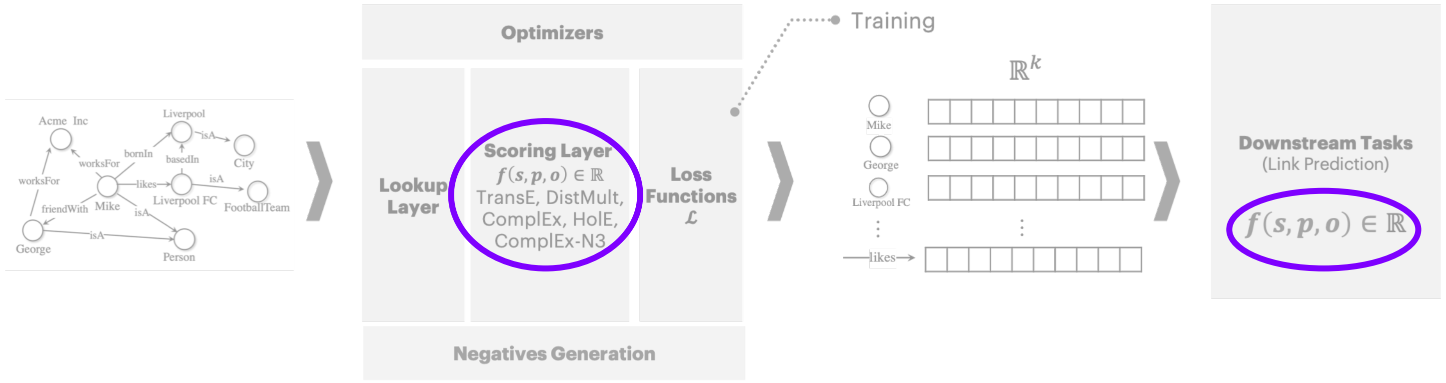
$\boldsymbol{f(s,p,o)} \in \mathbb{R}$

Inference

Anatomy of a Knowledge Graph Embedding Model

- Knowledge Graph (KG) $\mathcal{G}$
- Scoring function for a triple $f(t)$
- Loss function $\mathcal{L}$
- Optimization algorithm
- Negatives generation strategy

Scoring function f



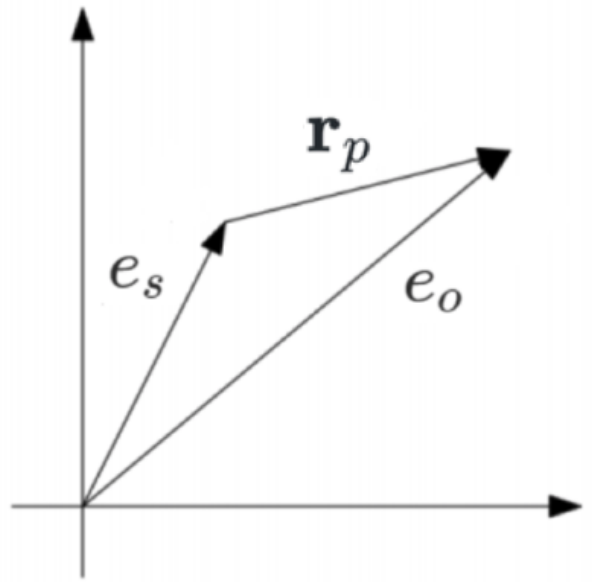
f assigns a score to a triple (s, p, o)

High score = high changes for the triple to be a true fact.

Translation-based Scoring Functions

- **TransE**: Translating Embeddings [Bordes et al. 2013]

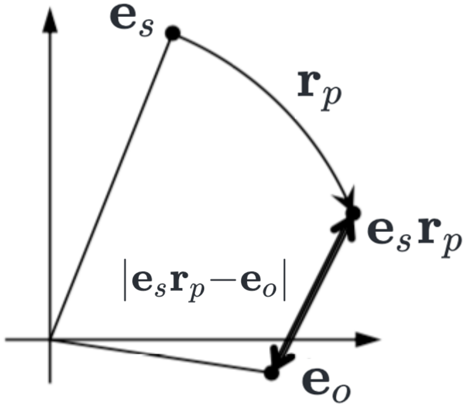
$$f_{TransE} = -||(\mathbf{e}_s + \mathbf{r}_p) - \mathbf{e}_o||_n$$



Translation-based Scoring Functions

- **RotatE**: relations modelled as *rotations* in complex space $\mathbb{C}$: element-wise product between complex embeddings. [Sun et al. 2019]

$$f_{RotatE} = -||\mathbf{e}_s \circ \mathbf{r}_p - \mathbf{e}_o||_n$$

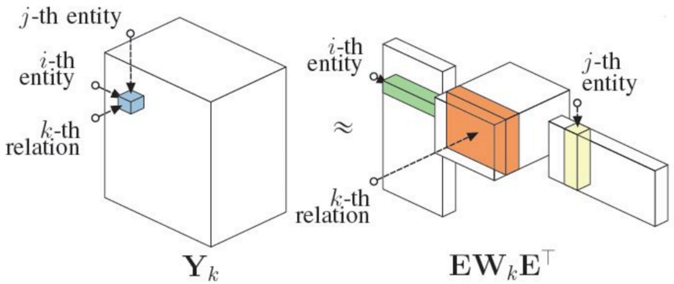


Factorization-based Scoring Functions

- **RESCAL**: low-rank factorization with tensor product

[Nickel et al. 2011]

$$f_{RESCAL} = \mathbf{e}_s^T \mathbf{W}_r \mathbf{e}_o$$



- **DistMult**: bilinear diagonal model. Dot product.

[Yang et al. 2015]

$$f_{DistMult} = \langle \mathbf{r}_p, \mathbf{e}_s, \mathbf{e}_o \rangle$$

- **ComplEx**: Complex Embeddings (Hermitian dot product):
  (i.e. extends DistMult with dot product in $\mathbb{C}$)

$$f_{ComplEx} = Re(\langle \mathbf{r}_p, \mathbf{e}_s, \overline{\mathbf{e}_o} \rangle)$$

[Trouillon et al. 2016]

“Deeper” Scoring Functions

- **ConvE**: reshaping + convolution

[Dettmers et al. 2017]

Non-linearity

$$f_{ConvE} = \langle \sigma \left(vec \left(g \left([\overline{\mathbf{e}_s}; \overline{\mathbf{r}_p}] * \Omega \right) \right) \mathbf{W} \right) \mathbf{e}_o \rangle$$

2D reshaping

Linear convolution

- **ConvKB**: convolutions and dot product

[Nguyen et al. 2018]

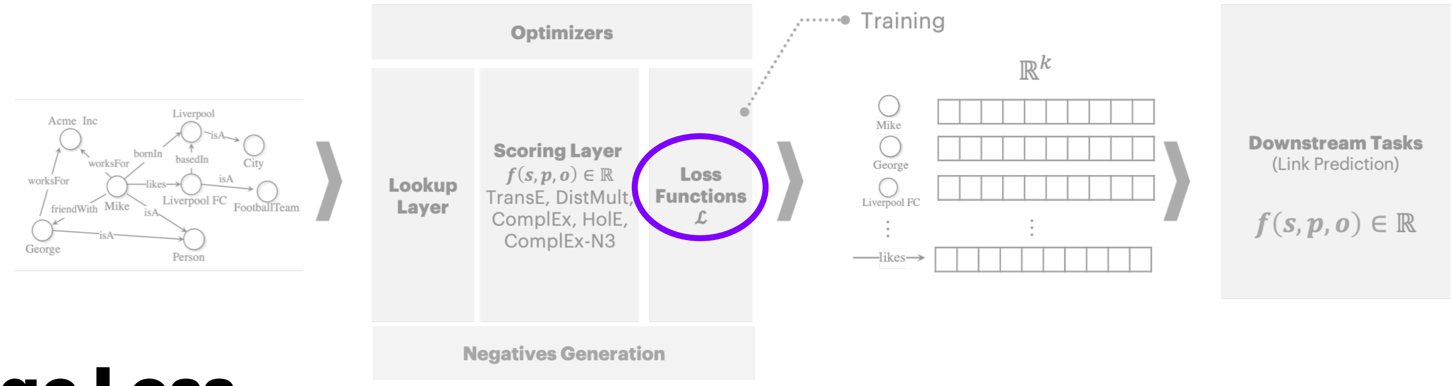
$$f_{ConvKB} = concat \left(g \left([\mathbf{e}_s, \mathbf{r}_p, \mathbf{e}_o] \right) * \Omega \right) \cdot W$$

Computationally expensive!

Other Recent Models

- HolE [Nickel et al. 2016]
- SimplE [Kazemi et al. 2018]
- QuatE [Zhang et al. 2019]
- MurP [Balažević et al. 2019]
- …

Loss function $\mathcal{L}$



Pairwise Margin-Based Hinge Loss

$$\mathcal{L}(\Theta) = \sum_{t^+ \in \mathcal{G}} \sum_{t^- \in \mathcal{C}} \overbrace{max(0, [\gamma + \underbrace{f(t^-;\Theta)}_{\text{Score assigned to a synthetic negative}} - \underbrace{f(t^+;\Theta)}_{\text{Score assigned to true triple}}])}^{\text{Pays a penalty if score of positive triple < score of synthetic negative by a margin } \gamma}$$

[Bordes et al. 2013]

Negative Log-Likelihood / Cross Entropy

$$\mathcal{L}(\Theta) = \sum_{t \in \mathcal{G} \cup \mathcal{C}} log(1 + exp(-\underbrace{y}_{\text{Label of the triple t}}\, f(t;\Theta)))$$

$y \in \{-1, 1\}$

[Trouillon et al. 2016]

Loss function $\mathcal{L}$

Binary Cross-Entropy

$$\mathcal{L} = -\frac{1}{N} \sum_{t \in \mathcal{G} \cup \mathcal{C}}^{N} y \cdot log(\sigma(f(t;\Theta))) + (1-y) \cdot log(1 - f(t;\Theta))$$

[Dettmers et al. 2017]

Self-Adversarial

$$\mathcal{L} = -log\,\sigma(\gamma + f(t^{+};\Theta)) - \sum_{t \in \mathcal{G}}^{N} \underbrace{p(t^{-};\Theta)}_{\text{Weight for the negative sample } t^{-}} log\,\sigma(-f(t^{-};\Theta) - \gamma)$$

[Sun et al. 2019]

Many more: Multiclass Negative Log-likelihood, Absolute Margin, etc.

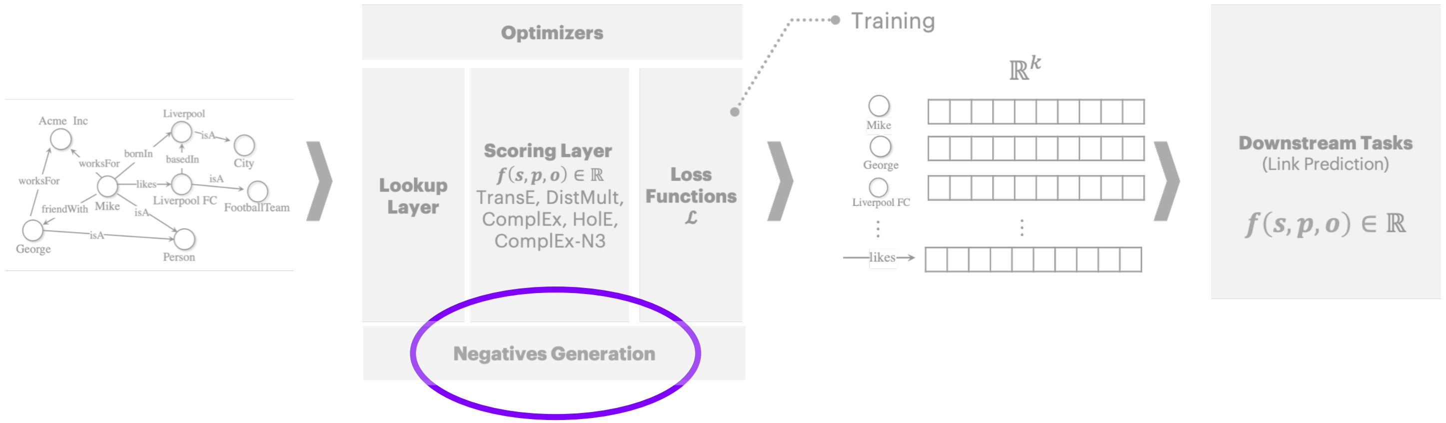
Regularizers

- L1, L2
- L3 [Lacroix et al. 2018]
- Dropout (ConvE) [Dettmers et al. 2017]

Initialization

- Random (Uniform)
- Random (Normal)
- Glorot

Negatives Generation



Where do negative examples come from? (i.e. false facts)

Local Closed World Assumption: the KG is only *locally* complete

“Corrupted” versions of a triple as synthetic negatives:

$$\mathcal{C} = \{(\hat{s}, p, o) | \hat{s} \in \mathcal{E}\} \cup \{(s, p, \hat{o}) | \hat{o} \in \mathcal{E}\}$$

“corrupted subject”

“corrupted” object

The predicate is unaltered

Synthetic Negatives: Example

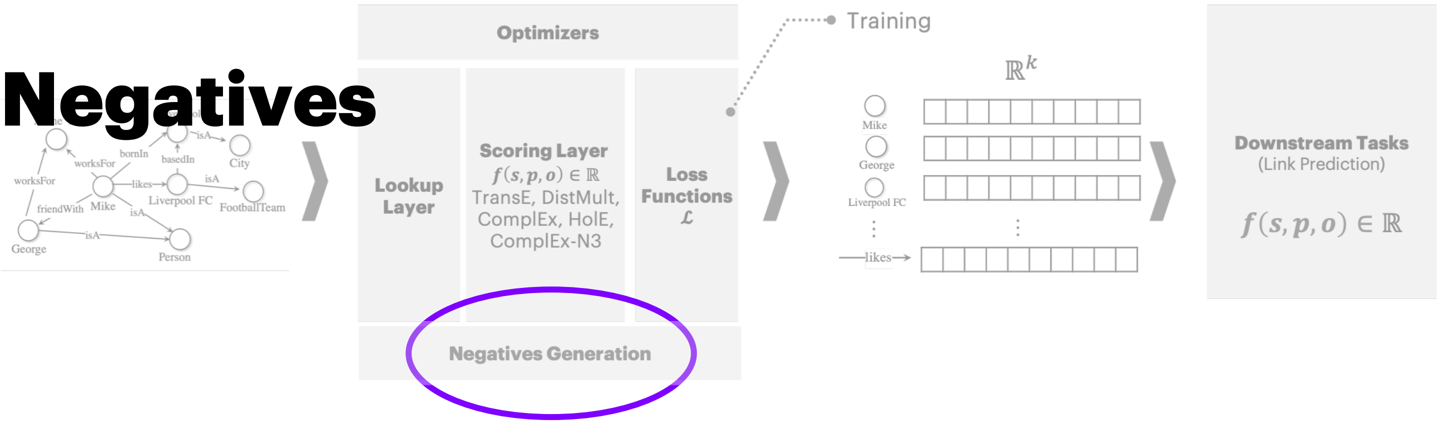
$$\mathcal{E} = \{Mike, Liverpool, AcmeInc, George, LiverpoolFC\}$$
$$\mathcal{R} = \{bornIn, friendWith\}$$

$t \in \mathcal{G} =$ `(Mike bornIn Liverpool)`

$\mathcal{C}_t =$

```
Mike      bornIn   AcmeInc
Mike      bornIn   LiverpoolFC
George    bornIn   Liverpool
AcmeInc   bornIn   Liverpool
```

Training with Synthetic Negatives

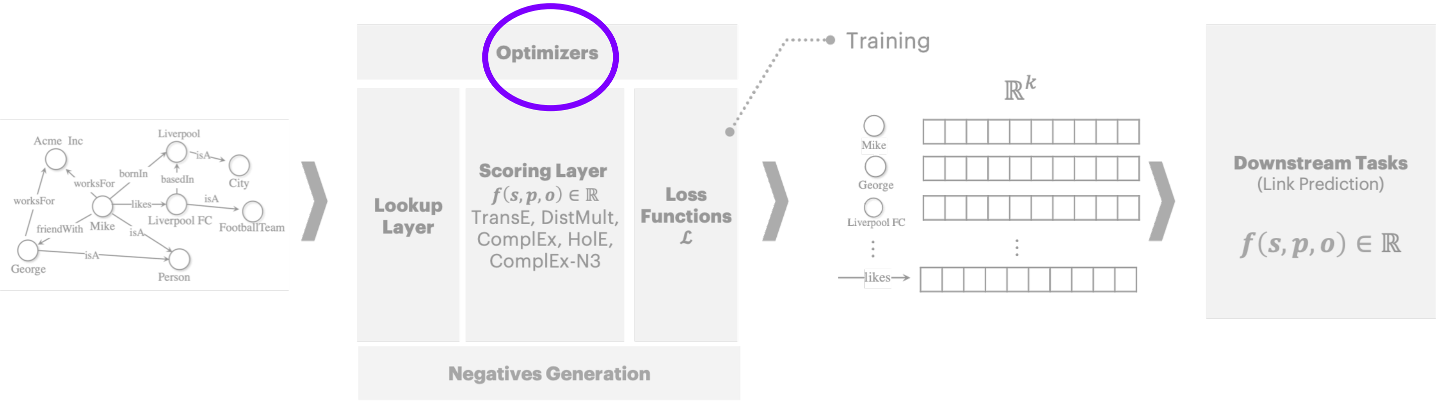


Uniform sampling: generate all possible synthetic negatives and sample n negatives for each positive t.

Complete set: no sampling. Use all possible synthetic negatives for each positive t. (mind scalability)

1-n scoring: batches of (s, p, *) or (*, p, o) labeled as positives (if included in training KG) or negatives (if not in training KG).
[Dettmers et al. 2017]

Training Procedure and Optimizer



Optimizer: learn optimal parameters (e.g. embeddings). Off-the-shelf SGD variants: (AdaGrad, Adam)

$$\min_{\Theta} \mathcal{L}(\Theta)$$

Reciprocal Triples

Injection of reciprocal triples in training set.

`<Alice childOf Jack>`

<Jack childOf^{-1} Alice>

[Dettmers et al. 2017]
[Lacroix et al. 2018]

Model Selection

- Grid search
  - Mind the size of the grid!
  - Early stopping
- Random search
- Quasi-random + Bayesian [Ruffinelli et a. 2020]

Outline

Theoretical Overview — 1h 30m

- Introduction
- Anatomy of a Knowledge Graph Embedding Model
- **Evaluation Protocol and Metrics**
- Advanced KGE Topics
- Open Research Questions

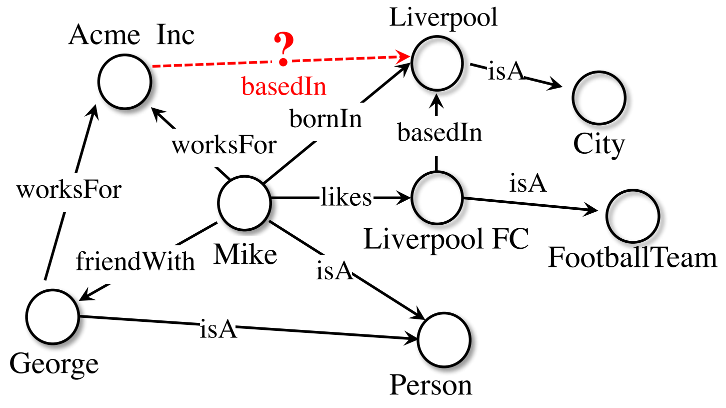
Applications — 15 m

Software Ecosystem — 15 m

Hands-on Sessions — 1h 15m

The Task

LINK PREDICTION / **TRIPLE CLASSIFICATION**



Assigning a score proportional to the likelihood that an unseen triple is true.

Link Prediction

- Learning to rank problem
- Information retrieval metrics
- No ground truth negatives in test set required

Triple Classification

- Binary classification task
- Binary classification metrics
- Test set requires positives and ground truth negatives

Same procedure used in training

Learning–To-Rank problem:

How well are positive triples ranked against **synthetic negatives** built under the **Local Closed World Assumption.**

Evaluation Metrics

Mean Rank (MR)

$$MR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} rank_{(s,p,o)_i}$$

Mean Reciprocal Rank (MRR)

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_{(s,p,o)_i}}$$

Hits@N

$$Hits@N = \sum_{i=1}^{|Q|} 1 \text{ if } rank_{(s,p,o)_i} \leq N$$

Example

How unseen, test positive triples rank against **synthetic negatives**?

s	p	o	score	rank	
Mike	born_in	Leeds	0.789	1	
Mike	born_in	Liverpool	0.753	2	*
Mike	born_in	Germany	0.695	3	
George	born_in	Liverpool	0.456	4	
Mike	born_in	George	0.234	5	

s	p	o	score	rank	
Mike	friend_with	George	0.901	1	*
Mike	friend_with	Jim	0.345	2	
Acme	friend_with	George	0.293	3	
Mike	friend_with	Liverpool	0.201	4	
France	friend_with	George	0.156	5	

Unseen positive triples (test set)

$$MR = 1.5$$
$$MRR = 0.75$$
$$Hits@1 = 0.5$$
$$Hits@3 = 1.0$$

Benchmark Datasets

Freebase

WordNet

YAGO

	FB15K-237	WN18RR	YAGO3-10
Training	272,115	86,835	1,079,040
Validation	17,535	3,034	5,000
Test	20,466	3,134	5,000
Entities	14,541	40,943	123,182
Relations	237	11	37

Link Prediction: SOTA Results

	FB15K-237	WN18RR	YAGO3-10
Literature Best	**0.35***	0.48*	0.49*
TransE (AmpliGraph)	0.31	0.22	**0.51**
DistMult (AmpliGraph)	0.31	0.47	0.50
ComplEx (AmpliGraph)	0.32	**0.51**	0.49
HolE (AmpliGraph)	0.31	0.47	0.50
ConvE (AmpliGraph)	0.26	0.45	0.30
ConvE (1-N, AmpliGraph)	0.32	0.48	0.40
ConvKB (AmpliGraph)	0.23	0.39	0.30

[https://github.com/Accenture/AmpliGraph]

		FB15K-237		*WNRR*	
		MRR	Hits@10	MRR	Hits@10
First	RESCAL (Wang et al., 2019)	27.0	42.7	42.0	44.7
	TransE (Nguyen et al., 2018)	29.4	46.5	22.6	50.1
	DistMult (Dettmers et al., 2018)	24.1	41.9	43.0	49.0
	ComplEx (Dettmers et al., 2018)	24.7	42.8	44.0	51.0
	ConvE (Dettmers et al., 2018)	32.5	50.1	43.0	52.0
Ours	RESCAL	**35.7**	**54.1**	46.7	51.7
	TransE	31.3	49.7	22.8	52.0
	DistMult	34.3	53.1	45.2	53.1
	ComplEx	34.8	53.6	**47.5**	**54.7**
	ConvE	33.9	52.1	44.2	50.4
Recent	TuckER (Balazevic et al., 2019)	**35.8**	**54.4**	47.0	52.6
	RotatE (Sun et al., 2019a)	33.8	53.3	**47.6**	**57.1**
	SACN (Shang et al., 2019)	35.0	54.0	47.0	54.4
Large	DistMult (Salehi et al., 2018)	35.7	54.8	45.5	54.4
	ComplEx-N3 (Lacroix et al., 2018)	**37.0**	**56.0**	**49.0**	**58.0**

[Ruffinelli et a. 2020]

Comparing SOTA Results is Tricky

- Different training strategies (e.g. synthetic negatives)
- Reciprocal relations in training set?
- Unfair or suboptimal hyperparameters selection
- Evaluation protocol: how to behave with tie ranks?
- Ablation studies!

Read discussion in [Ruffinelli et al 2020]

Outline

Theoretical Overview 1h 30m

- Introduction
- Anatomy of a Knowledge Graph Embedding Model
- Evaluation Protocol and Metrics
- **Advanced KGE Topics**
- Open Research Questions

Applications 15 m

Software Ecosystem 15 m

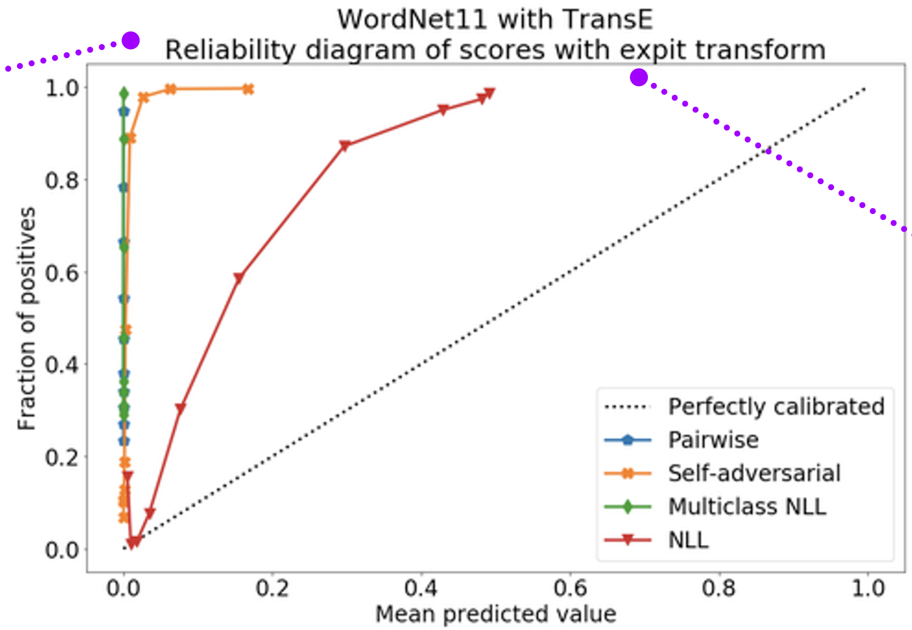
Hands-on Sessions 1h 15m

Calibration

Probabilities Generated by KGE models are Uncalibrated!

[Tabacof & Costabello 2020]

$f_m(\cdot) = 0.75$ confidence → model should be correct 75% of the times.



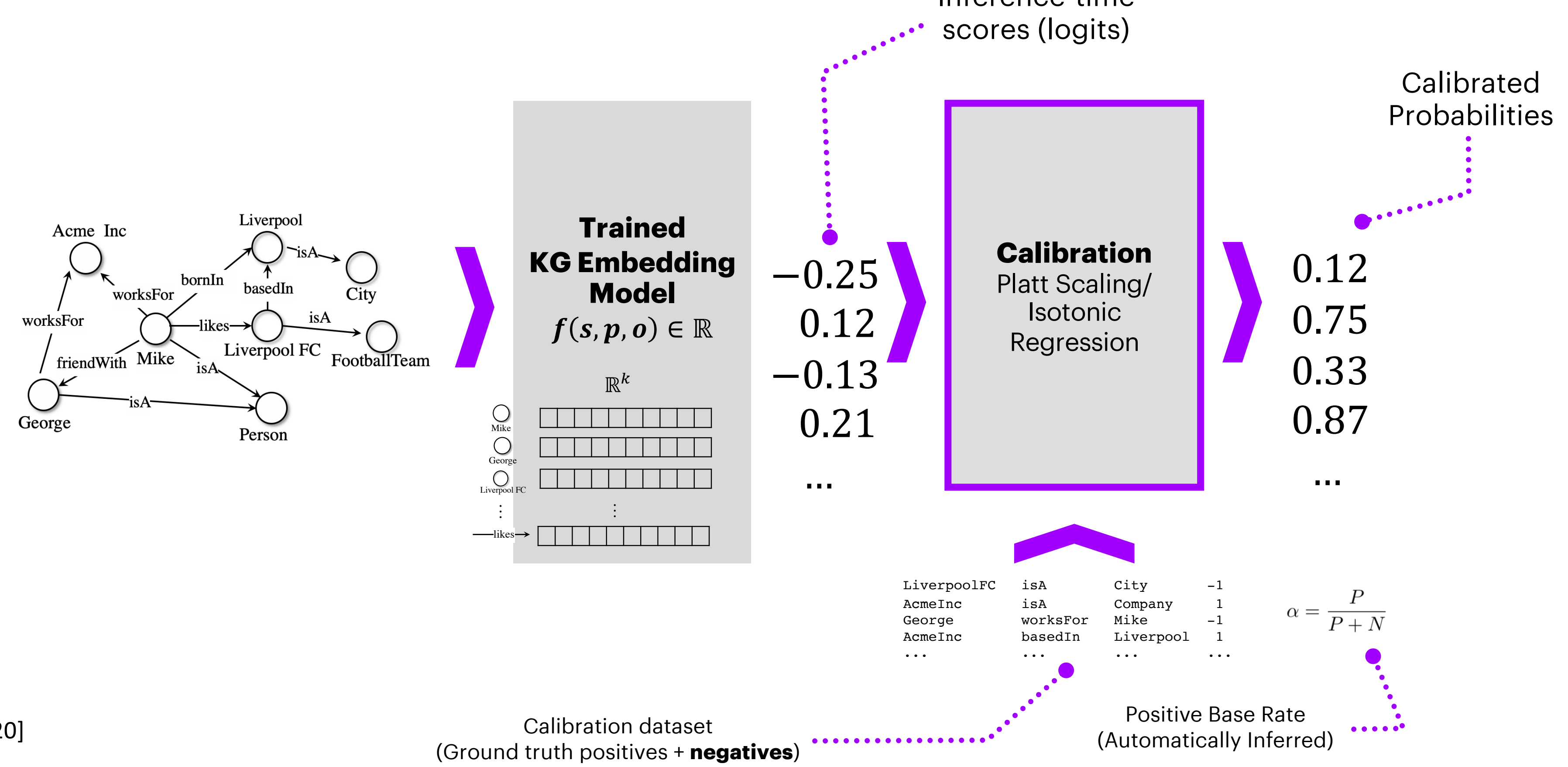
Logits + logistic sigmoid → poor calibration

- **Mistrust** in model discoveries
- **Poor Interpretability** in high-stakes scenarios (i.e. drug-target discovery)

How can we calibrate KGE models? How to do so without ground truth negatives?

Calibrating With **Ground Truth Negatives**

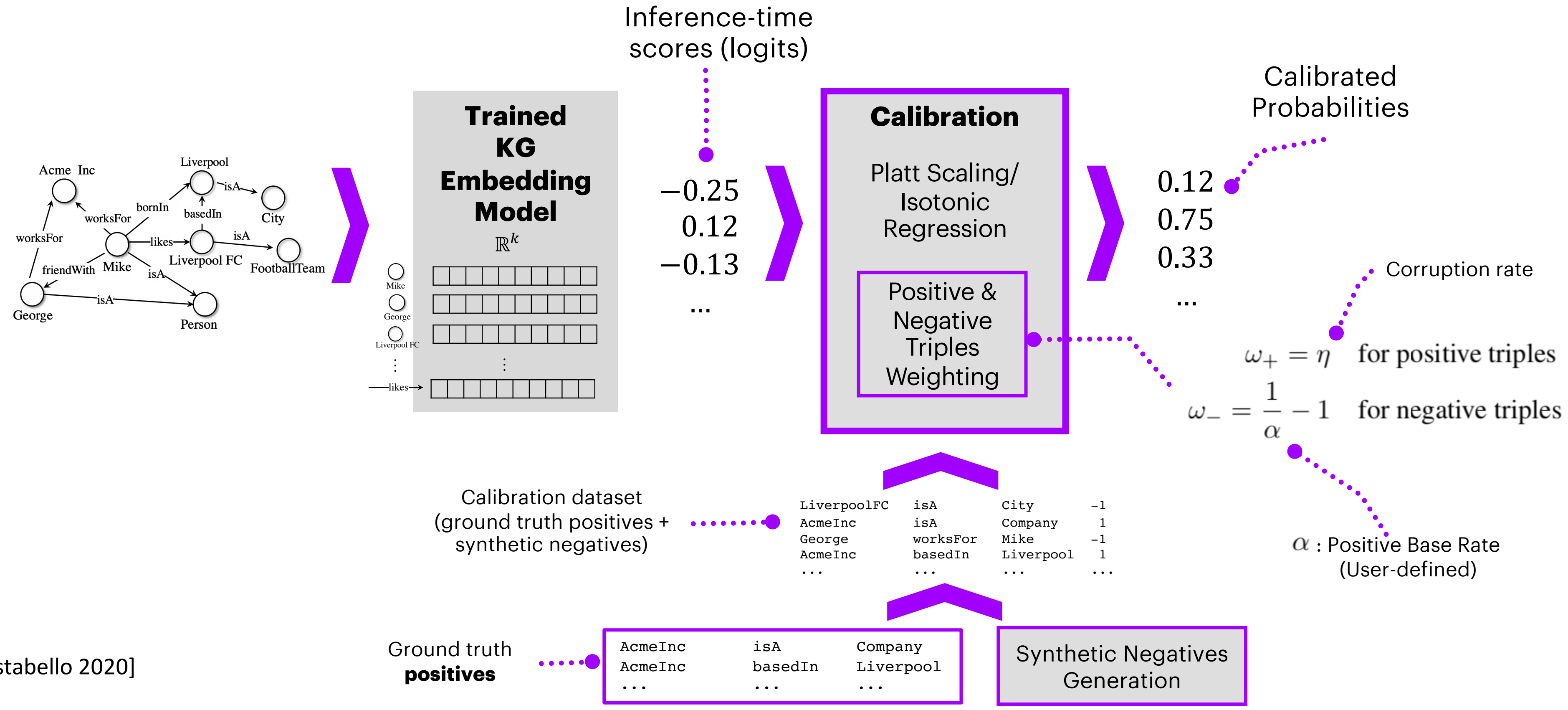
(Available w/ Triple Classification Datasets)



[Tabacof & Costabello 2020]

Calibrating With **Synthetic Negatives**

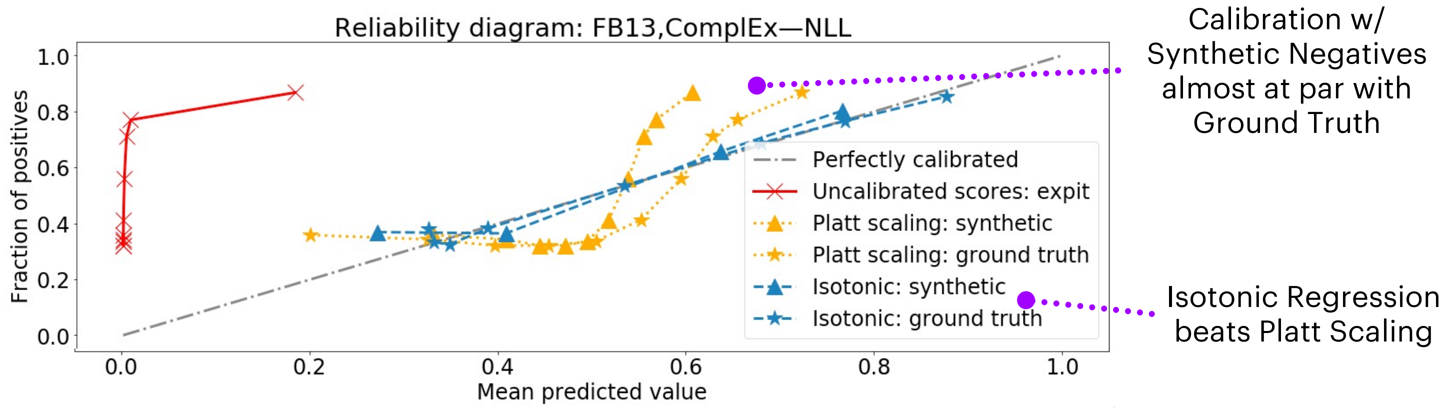
(No Ground Truth Negatives Available)



[Tabacof & Costabello 2020]

Calibration is Effective

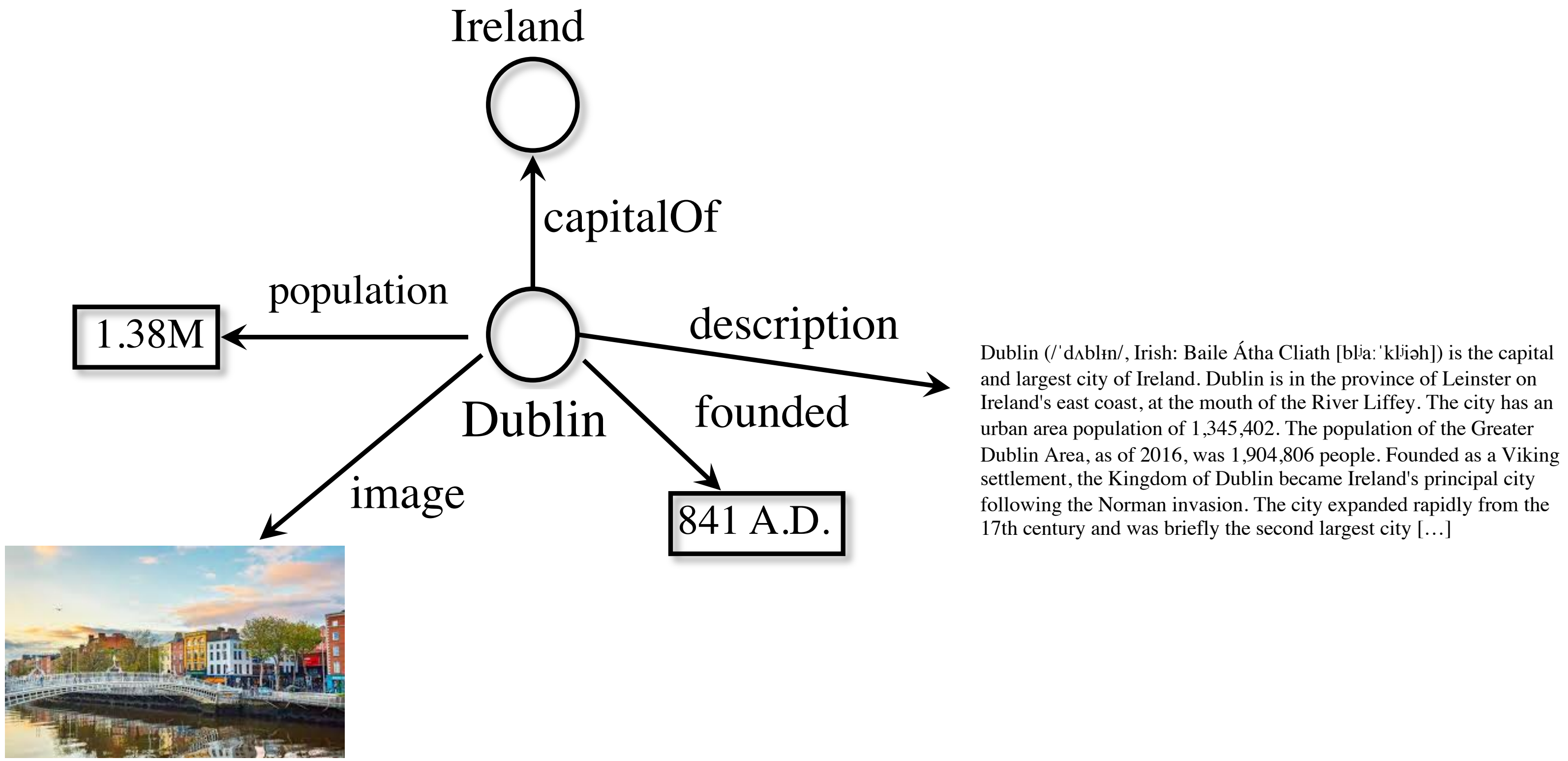
[Tabacof & Costabello 2020]



- All calibration techniques work **considerably better** than uncalibrated settings
- More **trustworthy and interpretable** predictions

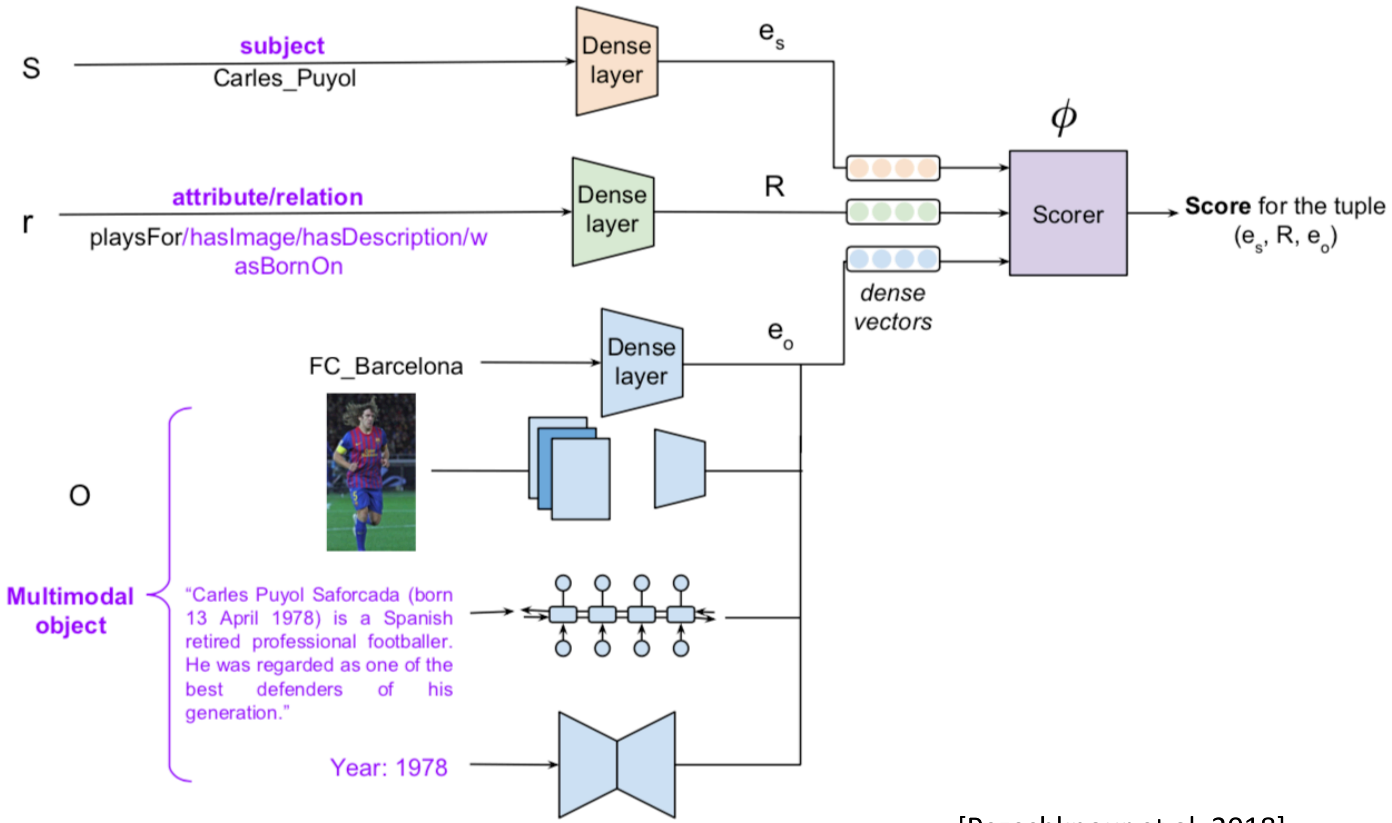
Multimodal Knowledge Graphs

Many real-world graphs includes **multi-modal attributes**.



Multimodal Knowledge Graph Embeddings

- KBLRN [Garcia-duran et al.2017]
- LiteralE [Kristiadi et al. 2018]
- MKBE [Pezeshkpour et al. 2018]

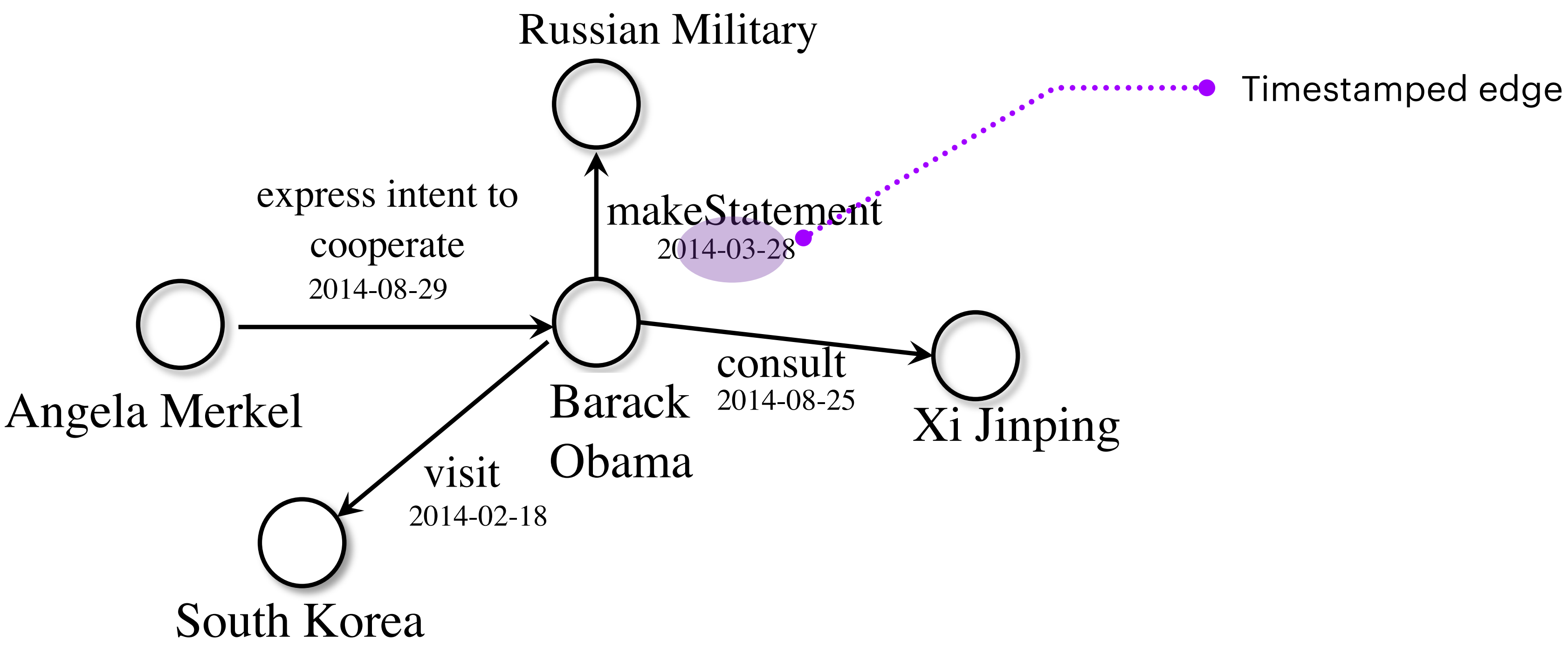


[Pezeshkpour et al. 2018]

[Gesese et al. 2019] surveys recent literature

Temporal Knowledge Graphs

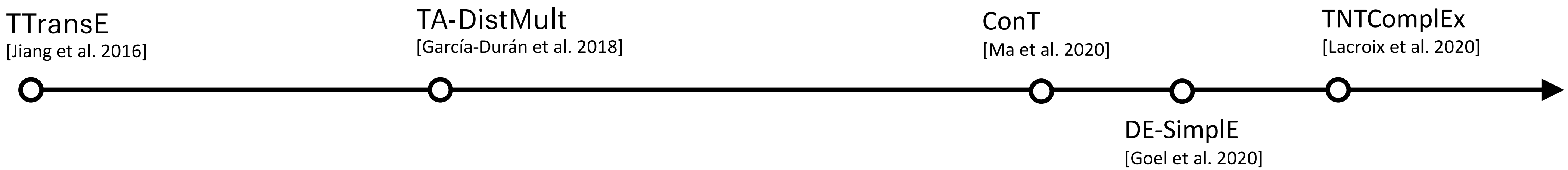
Many real-world graphs represents timestamped concepts.



	ICEWS14	ICEWS05-15	Yago15k	Wikidata
Entities	6869	10094	15403	432715
Predicates	460	502	102	814
Timestamps	365	4017	170	1726
|S|	72826	368962	110441	7224361

Table from [Lacroix et al. 2020]

Time Awareness: Temporal KGE models



TNTComplEx

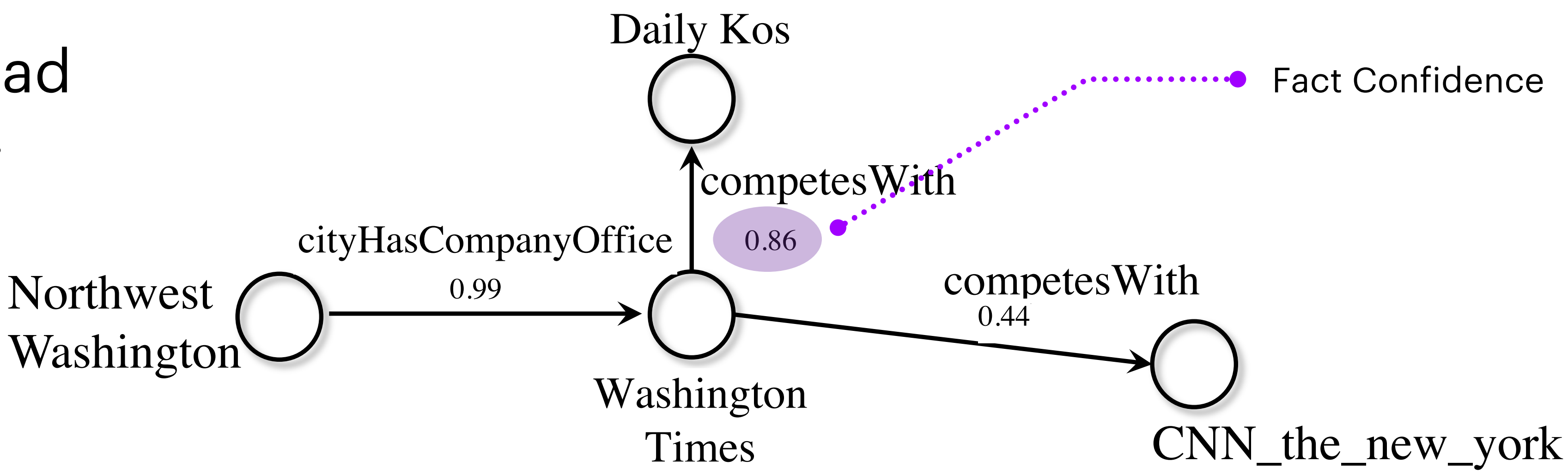
- Embeddings for each timestamp
- Order 4 tensor decomposition problem
- ComplEx as decomposition method

	ICEWS14	ICEWS15-05	Yago15k
TA	0.48	0.47	0.32
DE-SimplE	0.53	0.51	-
ComplEx	0.47 (0.47)	0.49 (0.49)	**0.35** (0.36)
TComplEx	**0.56** (0.61)	0.58 (0.66)	**0.35** (0.36)
TNTComplEx	**0.56** (**0.62**)	**0.60** (**0.67**)	**0.35** (**0.37**)

[Lacroix et al. 2020]

Uncertain Knowledge Graphs

Automatic KG generation may lead to *uncertain* facts.



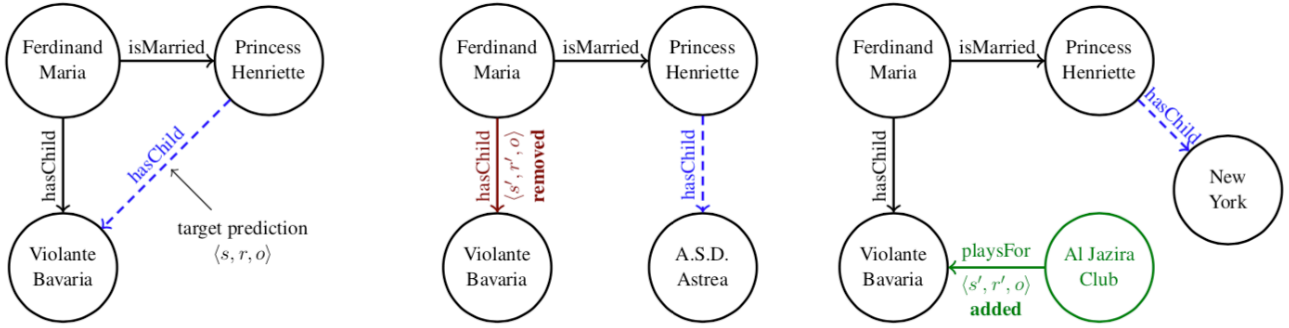
UKGE [Chen et al. 2019]

- Jointly training of KGE model + probabilistic soft logic to predict likelihood of unseen triples
- Logical rules are required as additional input

Robustness

KGE suffer from adversarial modifications

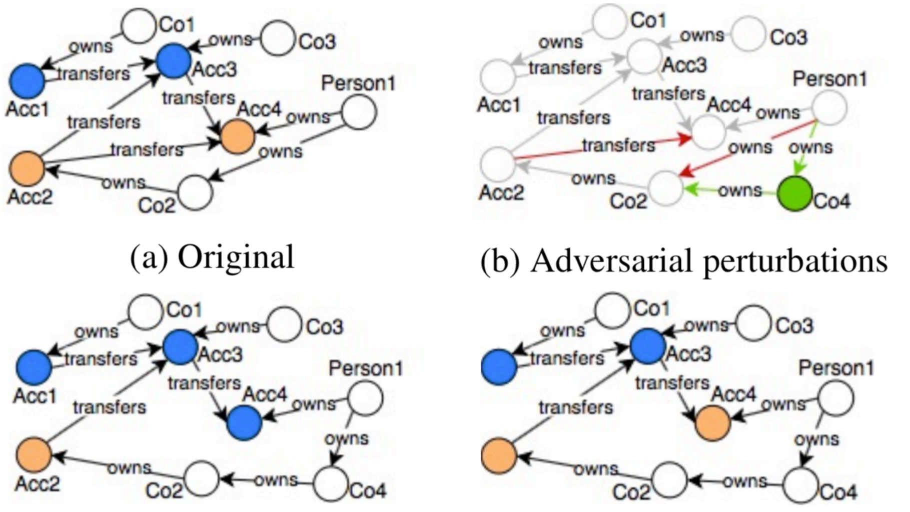
Link Prediction



(a) KG, with the target prediction (b) After removing a fact (c) After adding a fact

[Pezeshkpour et al. 2019]

Node Classification



(a) Original (b) Adversarial perturbations

(c) Adversarial attack (d) Adversarial robustness

[Bhardwaj 2020]

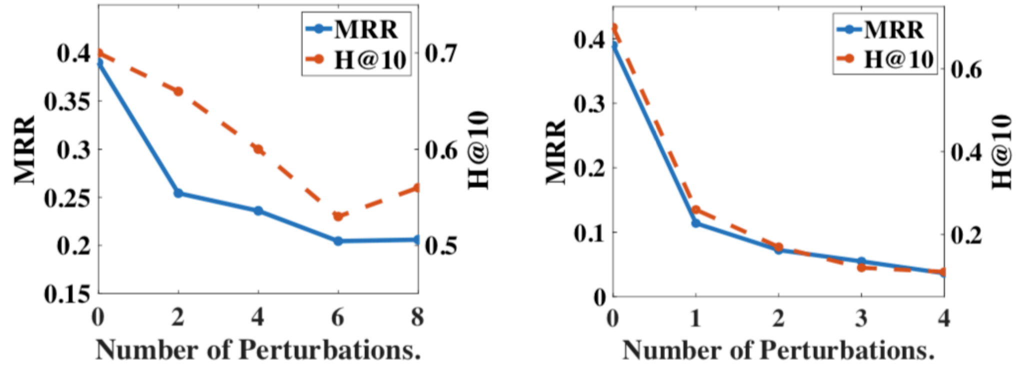
Robustness

Zhang et al. 2019

Generates input perturbations from the latent space by scoring all possible perturbations

CRIAGE [Pezeshkpour et al. 2019]

Encoder-decoder based inverter neural network



(a) Direct Adding Attack on WN18 Dataset against TransE

(b) Direct Deleting Attack on WN18 Dataset against TransE

KGE & Neuro-Symbolic Reasoning

Background knowledge injection with Soft Constraints

[Minervini et al. 2017]

Manually provide rules (or mine with AMIE+) and inject into loss function:

(a) Axioms extracted from YAGO3

hasNeighbor	$\equiv$	hasNeighbor^{-}
isMarriedTo	$\equiv$	isMarriedTo^{-}
playsFor	$\equiv$	isAffiliatedTo
isConnectedTo	$\equiv$	isConnectedTo^{-}

(b) Axioms extracted from DBpedia

assoc. band	$\equiv$	assoc. musical artist
musical band	$\equiv$	musical artist

We extend $\mathcal{L}$ with the **regularization term** $\mathcal{R}_{\mathcal{S}}$:

$$\mathcal{L}_{\mathcal{S}}(\Theta) = \mathcal{L}(\Theta) + \lambda \mathcal{R}_{\mathcal{S}}(\Theta)$$

$\lambda = \infty$ hard constraints

$\lambda = 0$ original model

$$\mathcal{R}_{\mathcal{S}}(\Theta) \triangleq \sum_{p \equiv q \in \mathcal{A}_1} D\left[\mathbf{r}_p \| \mathbf{r}_q\right] + \sum_{p \equiv q^{-} \in \mathcal{A}_2} D\left[\mathbf{r}_p \| \Phi(\mathbf{r}_q)\right]$$

$D[x||y] = ||x - y||_2^2$: *Divergence measure*

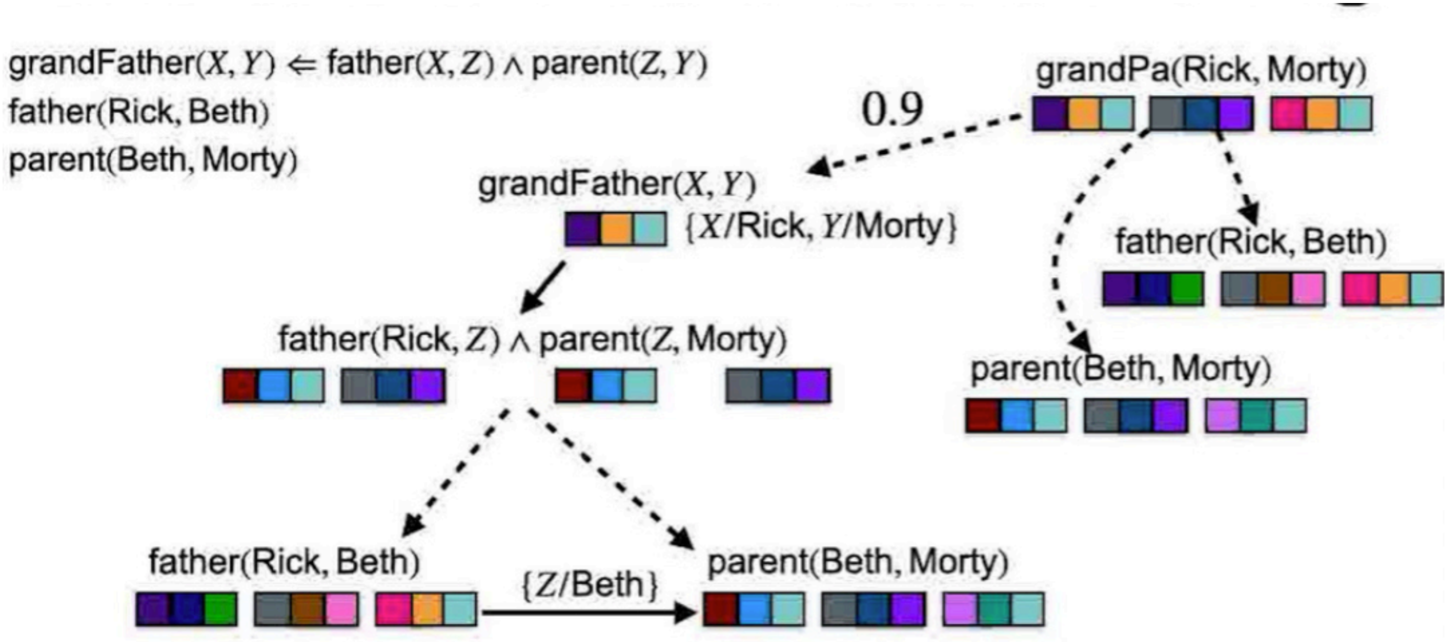
$\Phi(\cdot)$: *Model-dependent transformation*

$\mathcal{A}_1$: equivalent axioms set

$\mathcal{A}_2$: inverse axioms set

KGE & Neuro-Symbolic Reasoning: Neural Theorem Provers (NTP)

- Rule-based models + KGE
- Interplay of KGE strengths (good generalization power, scalability) with rule-based interpretability ("small data" capabilities).
- NTP implement reasoning (e.g. backward chaining) in fully differentiable architectures
    - Symbols replaced by embeddings
    - Compare embeddings in Prolog backward chaining instead of matching symbols

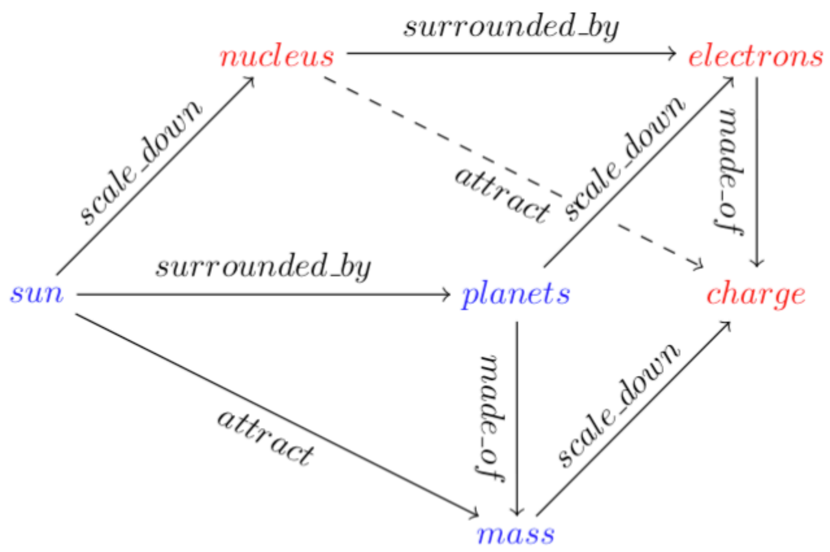


[Rocktäschel et al. 2017]
[Minervini et al. 2020]

Interplay with Other Reasoning Regimes: Analogical Reasoning

ANALOGY [Liu et al 2017]

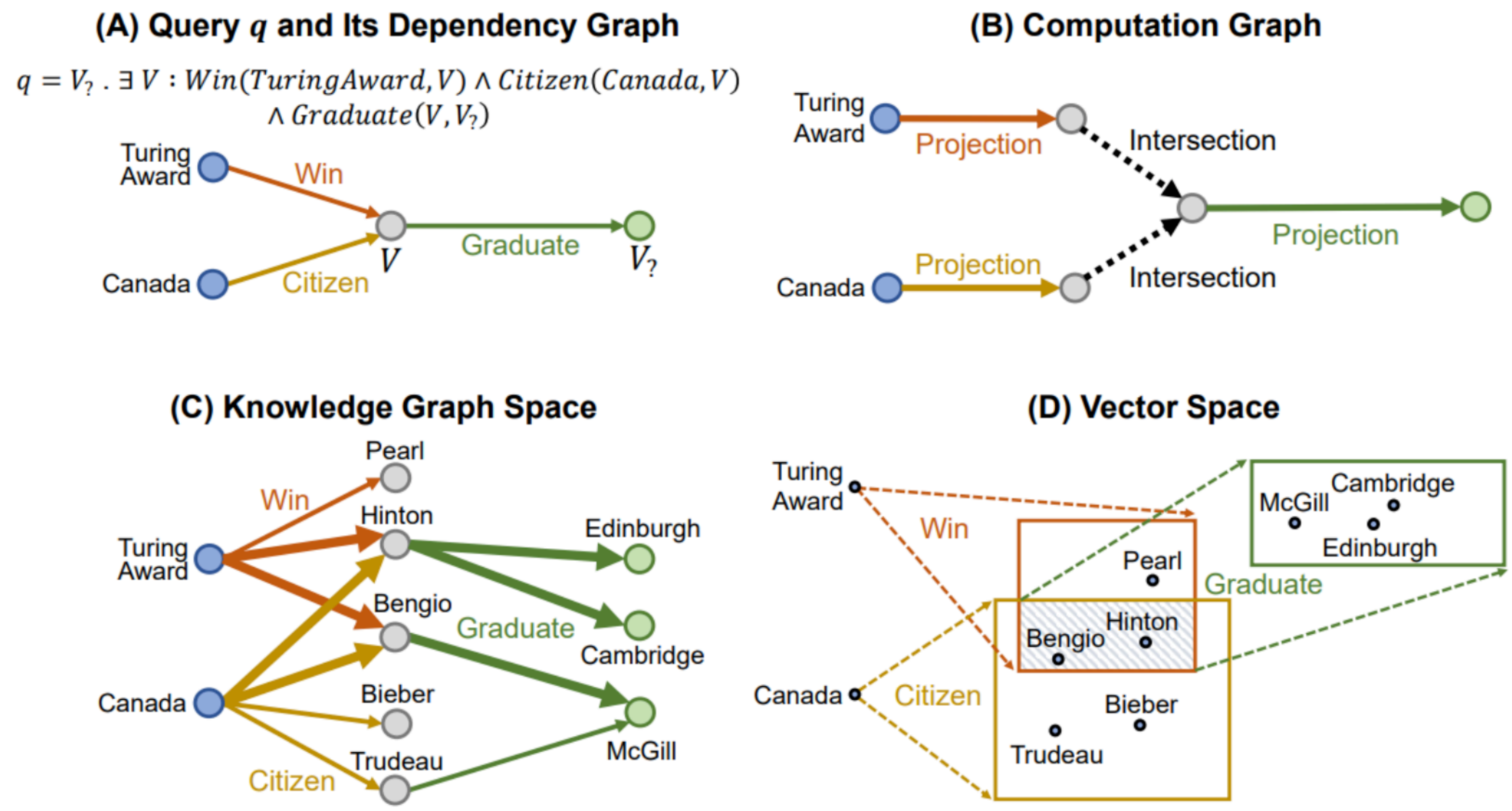
- Models analogical structures in multi-relational embeddings
- "Differentiable" analogical reasoning combined with KGE models



Answering Complex Queries

Query2box: reasoning over Knowledge Graphs in a vector space using *box embeddings* to answer complex queries.

[Ren et al. 2020]



Outline

Theoretical Overview — 1h 30m

- Introduction
- Anatomy of a Knowledge Graph Embedding Model
- Evaluation Protocol and Metrics
- Advanced KGE Topics
- **Open Research Questions**

Applications — 15 m

Software Ecosystem — 15 m

Hands-on Sessions — 1h 15m

Open Research Questions

MORE EXPRESSIVE MODELS

Model KG regularities and dependencies while keeping runtime/space complexity low.

SUPPORT FOR MULTIMODALITY

Node and edge attributes, time-awareness still in their infancy.

ROBUSTNESS & INTERPRETABILITY

Techniques to dissect, investigate, explain, and protect from adversarial attacks.

BETTER BENCHMARKS

Agreed-upon fair evaluation protocols, novel datasets.

BEYOND LINK PREDICTION

Multi-path predictions, adoption in larger differentiable architectures to inject background knowledge from graphs.

NEURO-SYMBOLIC INTEGRATION

Integrate KGE non-differentiable reasoning regimes to get the best of different worlds.

Outline

Theoretical Overview 1h 30m

- Introduction
- Anatomy of a Knowledge Graph Embedding Model
- Evaluation Protocol and Metrics
- Advanced KGE Topics
- Open Research Questions

Q&A

Applications 15 m

Software Ecosystem 15 m

Hands-on Sessions 1h 15m

References

[Auer et al. 2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: a nucleus for a web of open data. In The semantic web, 722–735. Springer, 2007.

[Bhardwaj 2020] Peru Bhardwaj, Towards Adversarially Robust Knowledge Graph Embeddings. AAAI 2020.

[Bianchi et al. 2020] Federico Bianchi, Gaetano Rossiello, Luca Costabello, Matteo Palmonari, and Pasquale Minervini. "Knowledge Graph Embeddings and Explainable AI." arXiv preprint arXiv:2004.14843 (2020).

[Bollacker et al. 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 1247–1250. AcM, 2008.

[Bordes et al. 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. NIPS, 2013.

[Chen et al. 2019] Chen, Xuelu, Muhao Chen, Weijia Shi, Yizhou Sun, and Carlo Zaniolo. "Embedding uncertain knowledge graphs." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 3363-3370. 2019.

[Danqi et al. 2015] Danqi and Chen. Observed versus latent features for knowledge base and text inference. In 3rd Workshop on Continuous Vector Space Models and Their Compositionality. ACL, 2015.

[Dettmers et al. 2017] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. AAAI, 2018

[García-Durán et al. 2017] Garcia-Duran, Alberto, and Mathias Niepert. "Kblrn: End-to-end learning of knowledge base representations with latent, relational, and numerical features." arXiv preprint arXiv:1709.04676 (2017).

[García-Durán et al. 2018] Alberto García-Durán, Sebastijan Dumancˇić, and Mathias Niepert. Learning sequence encoders for temporal knowledge graph completion. arXiv preprint arXiv:1809.03202, 2018.

[Gesese et al. 2019] Gesese, Genet Asefa, Russa Biswas, Mehwish Alam, and Harald Sack. "A Survey on Knowledge Graph Embeddings with Literals: Which model links better Literal-ly?." arXiv preprint arXiv:1910.12507, 2019.

[Glorot & Bengio 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, 249–256. 2010.

[Goodfellow et al. 2016] Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. Deep learning. Vol. 1. Cambridge: MIT press, 2016.

[Hamilton & Tang, 2019] William Hamilton, Jian Tang. Tutorial on Graph Representation Learning, AAAI, 2019.

[Hamilton 2020] William L. Hamilton. (2020). Graph Representation Learning. Morgan & Claypool, forthcoming .

[Hogan et al. 2020] Hogan, Aidan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo et al. "Knowledge graphs." arXiv preprint arXiv:2003.02320 (2020).

[Kadlec 2017] Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. Knowledge base completion: baselines strike back. CoRR, 2017.

[Kristiadi et al. 2018] Kristiadi, Agustinus, Mohammad Asif Khan, Denis Lukovnikov, Jens Lehmann, and Asja Fischer. "Incorporating literals into knowledge graph embeddings." arXiv preprint arXiv:1802.00934 (2018).

[Lacroix et al. 2018] Timothee Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. ICML, 2018.

[Li et al. 2018] Lisha Li and Kevin Jamieson. Hyperband: a novel bandit-based approach to hyperparameter optimization. Journal of Machine Learning Research, 18:1–52, 2018.

[Mahdisoltani et al. 2013] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. Yago3: a knowledge base from multilingual wikipedias. In CIDR. 2013.

[Ma et al 2018] Embedding models for episodic knowledge graphs. Journal of Web Semantics, 2018.

[Miller 1995] George A Miller. Wordnet: a lexical database for english. Communications of the ACM, 38(11):39–41, 1995.

[Minervini 2017] Minervini, Pasquale, Luca Costabello, Emir Muñoz, Vít Nováček, and Pierre-Yves Vandenbussche. "Regularizing knowledge graph embeddings via equivalence and inversion axioms.". ECML, 2017.

[Minervini 2020] Minervini, Pasquale, Sebastian Riedel, Pontus Stenetorp, Edward Grefenstette, and Tim Rocktäschel. "Learning Reasoning Strategies in End-to-End Differentiable Proving." ICML, 2020.

[Nguyen et al. 2018] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. NAACL-HLT, 2018.

[Nickel et al. 2016a] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. Procs of the IEEE, 104(1):11–33, 2016.

[Nickel et al. 2016b] Maximilian Nickel, Lorenzo Rosasco, Tomaso A Poggio, and others. Holographic embeddings of knowledge graphs. AAAI, 1955–1961. 2016.

[Pezeshkpour et al 2019] Pezeshkpour, Pouya, Yifan Tian, and Sameer Singh. Investigating Robustness and Interpretability of Link Prediction via Adversarial Modifications. NAACL, 2019.

[Ren et al. 2020] Ren, Hongyu, Weihua Hu, and Jure Leskovec. "Query2box: Reasoning over knowledge graphs in vector space using box embeddings.", ICLR 2020

[Rocktäschel et al. 2017] Rocktäschel, Tim, and Sebastian Riedel. "End-to-end differentiable proving." In NIPS 2017.

[Ruffinelli et al 2020] Ruffinelli, Daniel, Samuel Broscheit, and Rainer Gemulla. "You CAN teach an old dog new tricks! on training knowledge graph embeddings." ICLR, 2020.

[Socher et al. 2013] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. NIPS, 2013.

[Suchanek et al. 2007] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. WWW, 2007.

[Sun et al. 2019] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: knowledge graph embedding by relational rotation in complex space. In ICLR, 2019.

[Tabacof & Costabello 2020] Pedro Tabacof and Luca Costabello. Probability Calibration for Knowledge Graph Embedding Models. In ICLR, 2020.

[Toutanova et al. 2015] Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. Representing text for joint embedding of text and knowledge bases. EMNLP, 2015.

[Trouillon et al. 2016] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. ICML, 2016.

[Yang et al. 2014] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint, 2014.

[Zhang et al. 2019] Data Poisoning Attack against Knowledge Graph Embedding, IJCAI 2019.

Outline

Theoretical Overview 1h 30m

Applications 15 m

Software Ecosystem 15 m

Hands-on Sessions 1h 15m

Industrial applications:

Pharmaceutical Industry:
Drug Side-effects Prediction

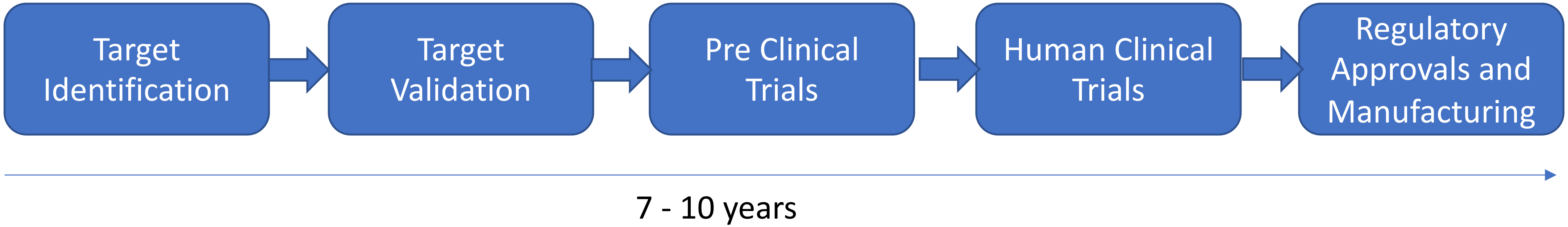
Human Resources:
Career Paths Prediction

Products:
Product Recommendation

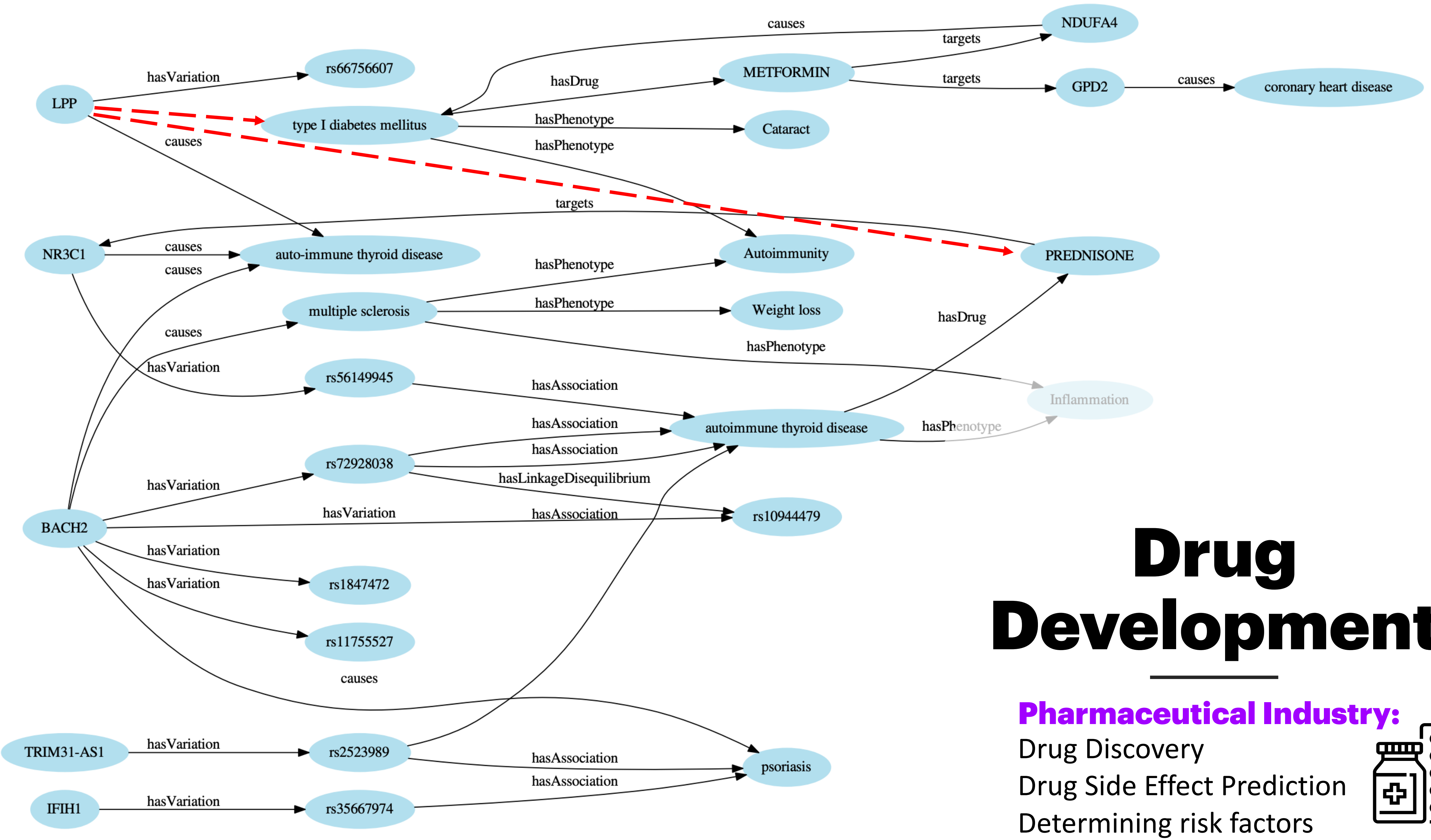
Food & Beverage:
Flavor Combinations

Drug Development

- Drug Development is a time consuming and expensive process which ranges from gene identification, identifying a compound to target the gene, and finally experimentation on animals and humans.



- The initial step of identification of gene/drug takes several years and if not identified correctly may result in loss of time and money.
- “Drug Developers” identify the genes/drugs by reading the latest research before proceeding with experimentation. But it is highly dependent on the experience of the person.



Drug Development

Pharmaceutical Industry:

Drug Discovery

Drug Side Effect Prediction

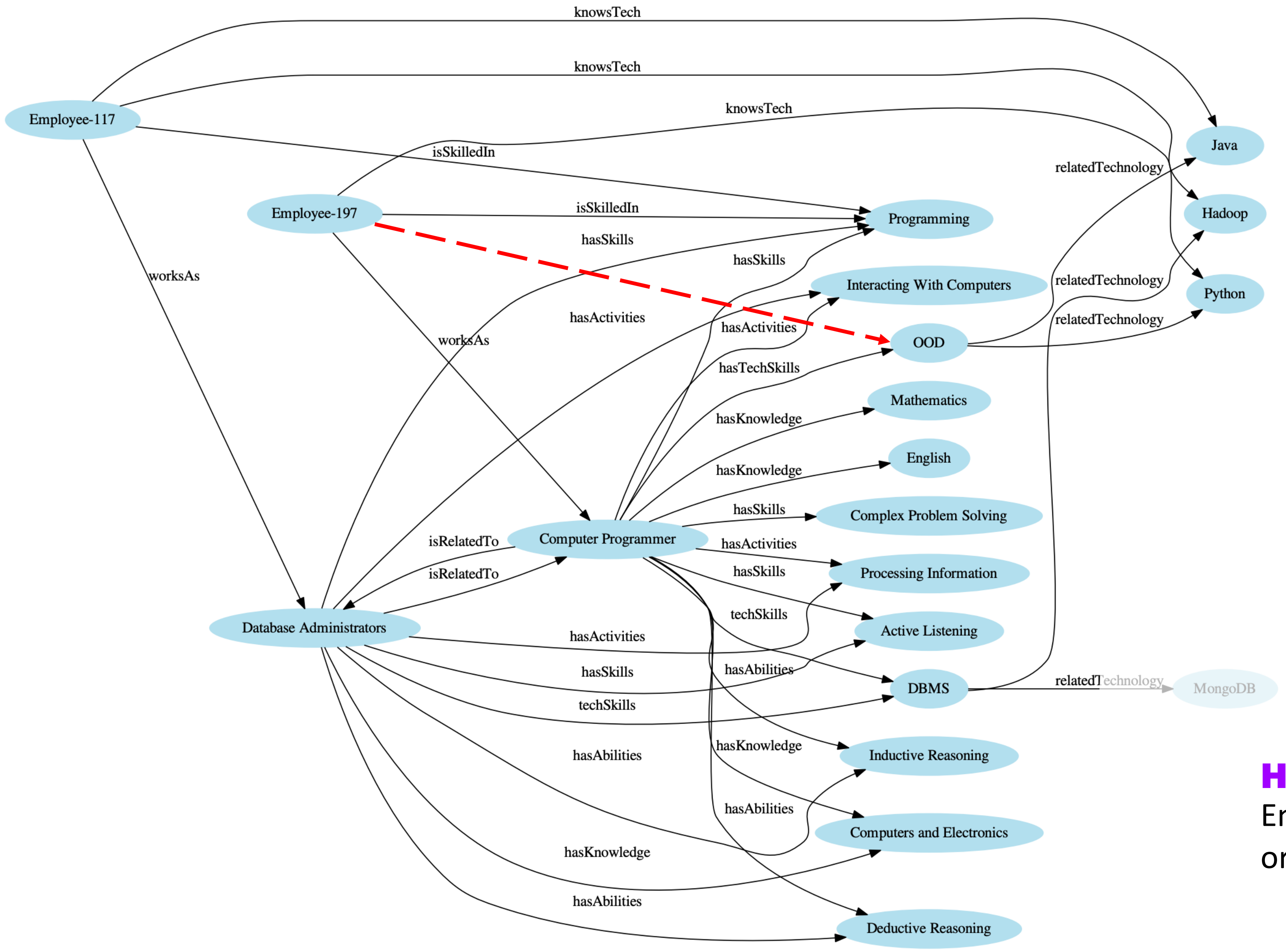
Determining risk factors

Human Resource

- Technology is evolving at an extremely fast pace. People need to learn new skills to be relevant in the market.
- Due to automation, a lot of roles are becoming obsolete and companies are forced to lay off people.

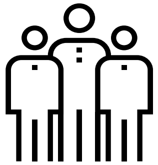
KGEs can be used for following tasks:

- Suggest new technology/tasks for career progression.
- Recommend similar roles within the organization when existing role becomes obsolete.



Human Resource

Human Resources:
Employee Career Progression or Transition

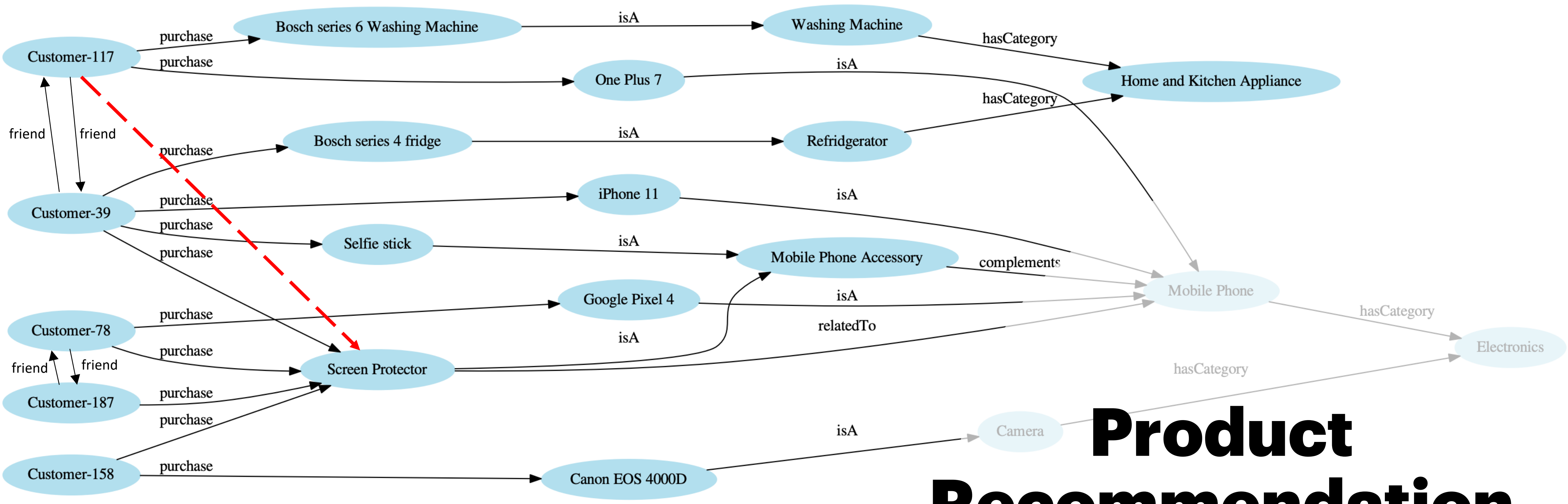


Product Recommendation

KGEs can leverage relation between customers and products.

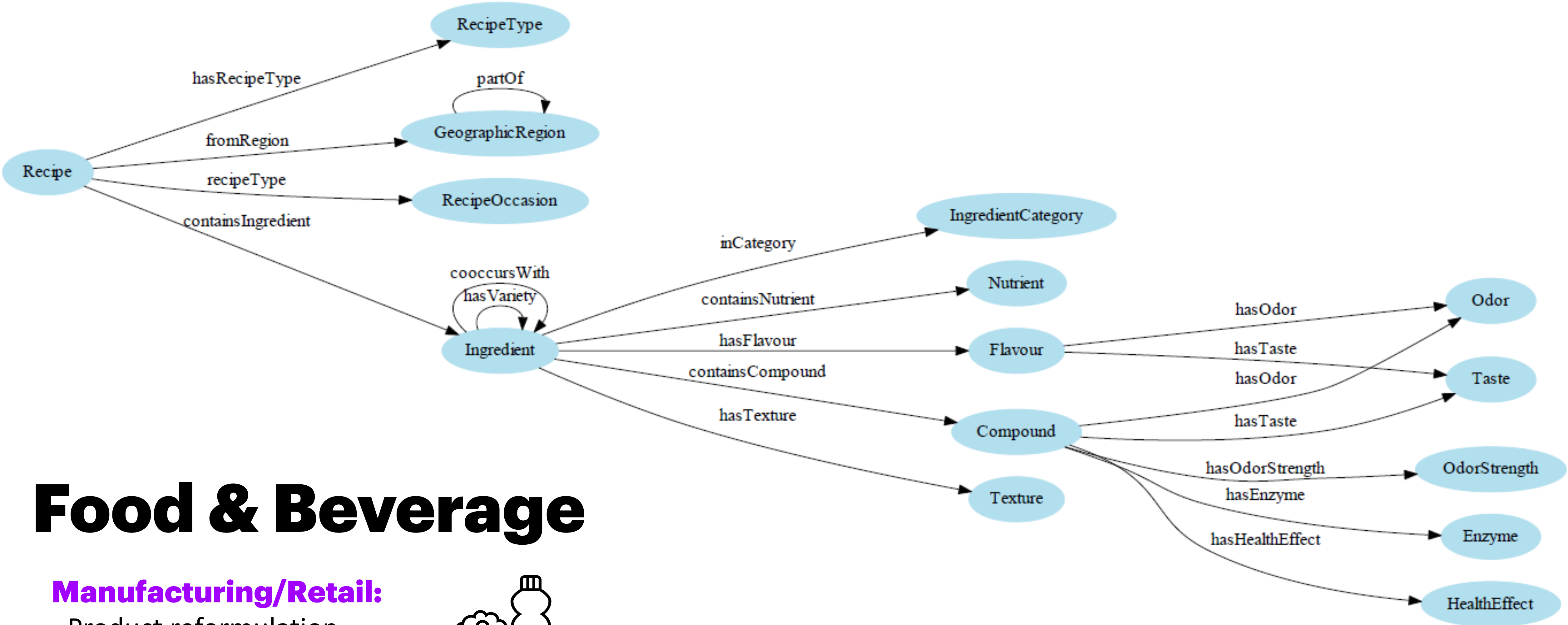
KGEs can be used for following tasks:

- Recommend new products to customers
- Group customers based on their purchase history



Product Recommendation

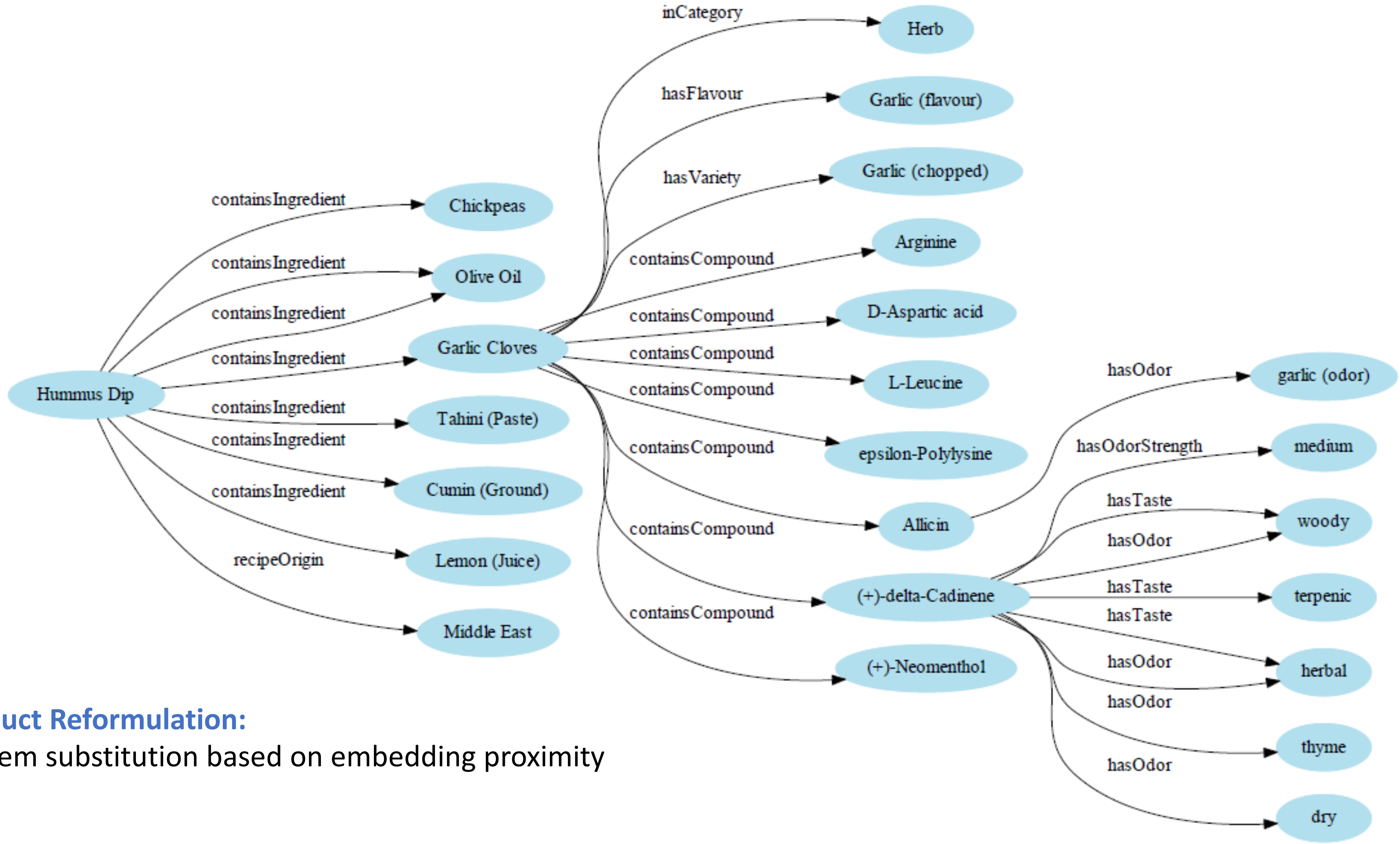
Retail:
Product Recommendation
Customer Grouping



Food & Beverage

Manufacturing/Retail:

- Product reformulation
- Adapting to consumer trends



Product Reformulation:

- Item substitution based on embedding proximity

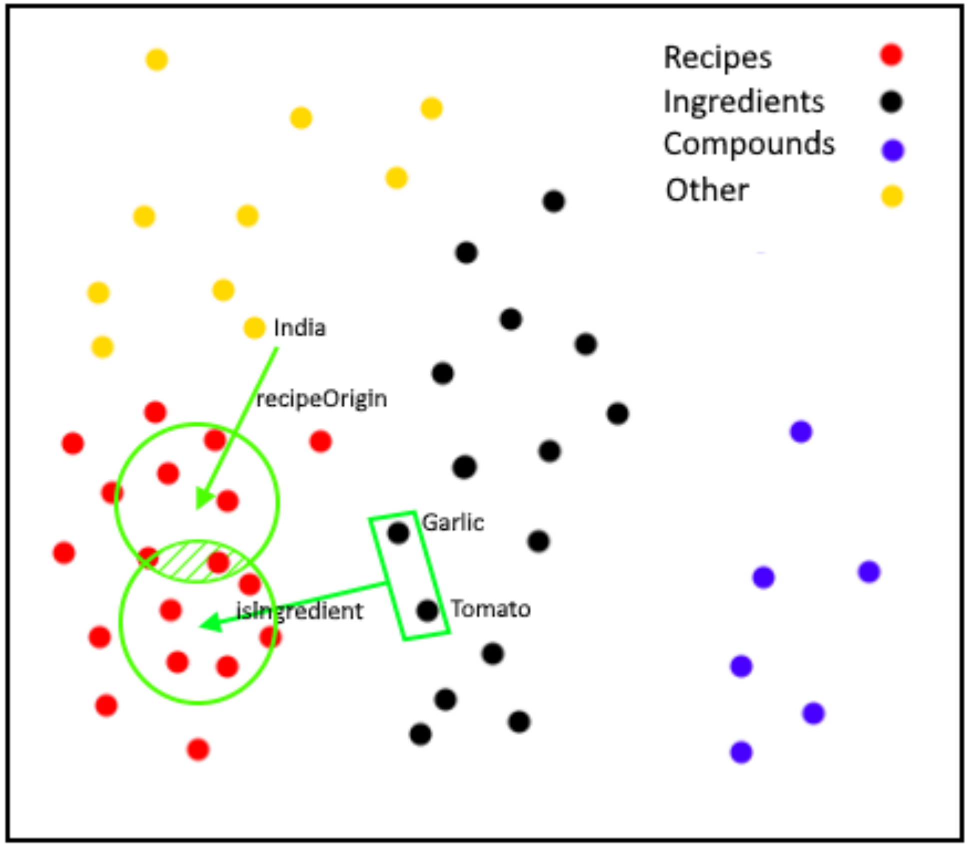
Item Recommendation

- Use vector algebra to find latent region that satisfy input criteria

- Example:
  - *"I want Indian recipes that contain garlic and tomato"*
  - $nearest(\,avg\,(avg(GARLIC, TOMATO) - containsIngredient, India\; - recipeOrigin))$

- *Note: the above is pseudo-code, actual solutions will depend on model, data, fine-tuning, etc*

- Alternatively use Bayesian optimization ..



Graph Construction

Datasets

Data processing

Pancakes, baking powder, eggs, all-purpose flour, raisins, milk, white sugar

Hummus, olive oil, chickpeas, lemon juice, tahini, garlic, cumin

...

Garlic, (E)-2-Phenyl-2-butenal, Arginine, D-Aspartic acid L-Leucine, epsilon-Polylysine, (±)-erythro-Isoleucine, Potassium

...

(+)-delta-Cadinene, 441005, herbal, woody, thyme, wood, medicine, dry
1,3-Dithiane, 10451, roasted, alliaceous

..

(+)-delta-Cadinene Molecular Weight, 204.35 XLogP3-AA, 3.8 Pathway, gossypol biosynthesis Pathway, lacinilene C biosynthesis

..

Triples

Recipe 1, contains, Garlic
Recipe 1, contains, Lemon juice
..

Garlic, hasCompound, Arginine
Garlic, hasCompound, L-Leucine
...

(+)-delta-Cadinene, hasOdor, herbal
(+)-delta-Cadinene, hasOdor, thyme
...

(+)-delta-Cadinene, hasPathway, gossypol
(+)-delta-Cadinene, hasPathway, lacinilene
...

Knowledge Graph

Further reading ..

- **BioKEEN: A library for learning and evaluating biological knowledge graph embeddings**. Ali, M., Hoyt, C. T., Domingo-Fernández, D., Lehmann, J., Jabeen, H., & Wren, J. (2019). *Bioinformatics*, *35*(18), 3538–3540.
- **Benchmark and Best Practices for Biomedical Knowledge Graph Embeddings**. Chang, D., Balažević, I., Allen, C., Chawla, D., Brandt, C., & Taylor, A. (2020). 167–176.
- **Knowledge Graph Embedding for Ecotoxicological Effect Prediction.** Myklebust, E. B., Jimenez-Ruiz, E., Chen, J., Wolf, R., & Tollefsen, K. E. (2019). *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11779 LNCS*, 490–506.
- **Metaresearch Recommendations using Knowledge Graph Embeddings**. Henk, V., Vahdati, S., Nayyeri, M., Ali, M., Yazdi, H. S., & Lehmann, J. (2019). *The AAAI-19 Workshop on Recommender Systems and Natural Language Processing (RecNLP).*
- **Product knowledge graph embedding for E-commerce.** Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., & Achan, K. (2020). *WSDM 2020 - Proceedings of the 13th International Conference on Web Search and Data Mining*, 672–680.
- ***Stock Price Movement Prediction from Financial News with Deep Learning and Knowledge Graph Embedding.*** Yang, W., B, S. G., Raza, A., Herbert, D., & Kang, B. (2018). *15th Pacific Rim Knowledge Acquisition Workshop* (Vol. 11016). Springer International Publishing.
- **Knowledge Graph-based Event Embedding Framework for Financial Quantitative Investments.** Dawei Cheng, Fangzhou Yang, Xiaoyang Wang, Ying Zhang, and Liqing Zhang. 2020.. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20).
- **Linking physicians to medical research results via knowledge graph embeddings and twitter.** Sadeghi, A., & Lehmann, J. (2020). *Communications in Computer and Information Science*, *1167 CCIS*, 622–630.
- **Knowledge graph embeddings with node2vec for item recommendation**. Palumbo, E., Rizzo, G., Troncy, R., Baralis, E., Osella, M., & Ferro, E. (2018). *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11155 LNCS*, 117–120.

Outline

Theoretical Overview — 1h 30m

- Introduction
- Anatomy of a Knowledge Graph Embedding Model
- Evaluation Protocol and Metrics
- Advanced KGE Topics
- Open Research Questions

Applications — 15 m

Software Ecosystem — 15 m

Hands-on Sessions — 1h 15m

Outline

Theoretical Overview

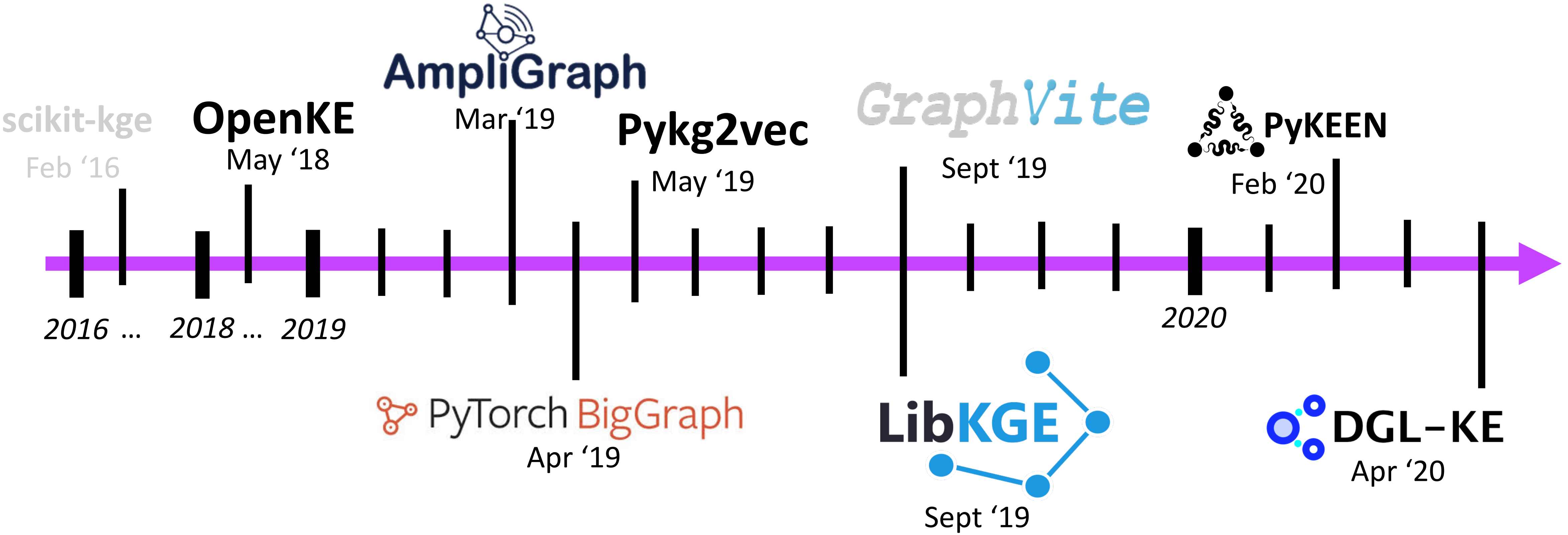
Applications

Software Ecosystem

- Introduction
- What is Out There?
- Libraries Comparison
  - Features
  - Scalability
  - SOTA Reproduced
  - Software Development
- Which Library Should I Use?
- Questions

Hands-on Sessions

KGE's Software Universe



Date reported is first of the following: pre-release/release/tag, in case where there are none of these the date reported is either submission date of a published paper accompanying or induced from repository activity. This is the case for scikit-kge, OpenKE, LibKGE.

Libraries comparison

… and what we measured

- Features
- Scalability
- SOTA Reproduced
- Software Development

Features

Models

Pre-trained models

Other Features

...

Models

	TransE	DistMult	ComplEx	TransH	TransD	TransR	RESCAL	HolE	SimplE	Analogy	ConvKB	ConvE
OpenKE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
AmpliGraph	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗	✓	✓
PyTorch BigGraph	✓	✓	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗
GraphVite	✓	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗
DGL-KE	✓	✓	✓	✗	✗	✓	✓	✗	✗	✗	✗	✗
PyKEEN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
Pykg2vec	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Lib-KGE	✓	✓	✓	✗	✗	✗	✓	✗	✓	✗	✗	✓
scikit-kge	✓	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗

	RotatE	QuatE	KG2E	NTN	ProjE	RGCN	TuckER	TransM	CP	Other models
OpenKE	✗	✗	✗	✗	✗	✗	✗	✗	✗	
AmpliGraph	✗	✗	✗	✗	✗	✗	✗	✗	✗	
PyTorch BigGraph	✗	✗	✗	✗	✗	✗	✗	✗	✗	
GraphVite	✓	✓	✗	✗	✗	✗	✗	✗	✗	
DGL-KE	✓	✗	✗	✗	✗	✗	✗	✗	✗	
PyKEEN	✓	✗	✓	✓	✓	✓	✓	✗	✗	Complex /DistMultLiteral, ERMLP, StructuredEmbedding, SME
Pykg2vec	✓	✗	✓	✓	✓	✗	✓	✓	✓	SLM, SME, ComplexN3
Lib-KGE	✓	✗	✗	✗	✗	✗	✓	✗	✓	
scikit-kge	✗	✗	✗	✗	✗	✗	✗	✗	✗	ERMLP

Pre-trained Models

	WikiData dump	Freebase	Benchmark datasets
OpenKE	✓ link	✓ (fragment ?)	✗
AmpliGraph	✗	✗	✓ (upon request)
PyTorch BigGraph	✓ (full)	✗	✗
GraphVite	✓ (Wikidata5m)	✗	✗
DGL-KE	✗	✗	✗
PyKEEN	✗	✗	✗
Pykg2vec	✗	✗	✗
Lib-KGE	✓ (Wikidata5M)	✗	✓ link
scikit-kge	✗	✗	✗

OpenKE requires submitting your name, email and organization before download.
? The size of this embedding suggest it is a fragment.

Other Features

- OpenKE
  - C++ implementation.
- AmpliGraph
  - Benchmarking Aid and pre-processing.
  - Formats: rdf, csv, ntriples.
  - Knowledge discovery API.
  - Visualization.
  - Model selection API.
  - Slack.
  - Colab Tutorials.
- PBG
  - High-level operators.
  - Scalability (partitioning, experimental GPU).
- DGL-KE
  - Scalability (partitioning with METIS, faster than GraphVite and PBG).
- PyKEEN
  - Incorporating multi-modal information.
  - Extensibility (wide range of interchangeable components).
  - Hyperparameters support (Optuna).
- Pykg2vec
  - Metrics summary plots.
  - Automatic discovery for hyperparameters.
  - Interactive results inspector.
- Lib-KGE
  - Hyper param support (includes Bayesian Optimization).
  - Resuming training.
  - Configuration via yaml.
- GraphVite
  - Command line interface.
  - Visualization.
  - Configuration via yaml.
  - Auto-deduction of hyperparameters.
  - Scalability (GPU-CPU hybrid).
  - Node Embedding API.
  - Input data parser.

**scikit-kge is not listed here as it was discontinued*

Scalability

Scalability

	Core Framework	GPU	Distributed Execution CPU	Biggest Graph
OpenKE	PyTorch/Tensorflow	✓	✗	10^8 edges, $4x10^7$ nodes [3]
AmpliGraph	Tensorflow	✓	✗ (Coming)	10^8 edges, 10^6 nodes [1]
PyTorch BigGraph	PyTorch	✓	✓	$2.4x10^{12}$ edges, $1.21x10^8$ nodes [4]
GraphVite	PyTorch	✓	✓ (GPU-CPU)	$1.8x10^{12}$ edges, $6.6x10^6$ nodes [9]
DGL-KE	PyTorch	✓	✓	$3.38x10^8$ edges, $8.6x10^6$ nodes [8]
PyKEEN	PyTorch	✓	✗	-
Pykg2vec	PyTorch/Tensorflow	✓	✗	-
Lib-KGE	PyTorch	✓	✗	-
scikit-kge	-	✗	✗	-

SOTA Reproduced

SOTA Reproduced

	OpenKE	PBG	AmpliGraph	GraphVite	DGL-KE	PyKEEN	Pykg2vec	Lib-KGE	scikit-kge
SOTA reproduced	✓	✓	✓	✓	✓	✗	✓	✓	✓
Models reported	8/10	2/4[1]	6/6	6/6	6/6	0/22[?]	10/22	9/9	5/5[2]

[1] page 8, Lerer et. al. 2019.
[2] page 6, Nickel et. al. 2015.
[?] Not found.

Software Development

Software Development Metrics

- Documentation (docstr-coverage) - PEP 257
  - Counts number of functions, classes, methods, and modules that doesn't have docstrings.
- Tests (coverage)
  - measures how many lines out of the executable lines were executed.
- Good practices (pylint) PEP 8

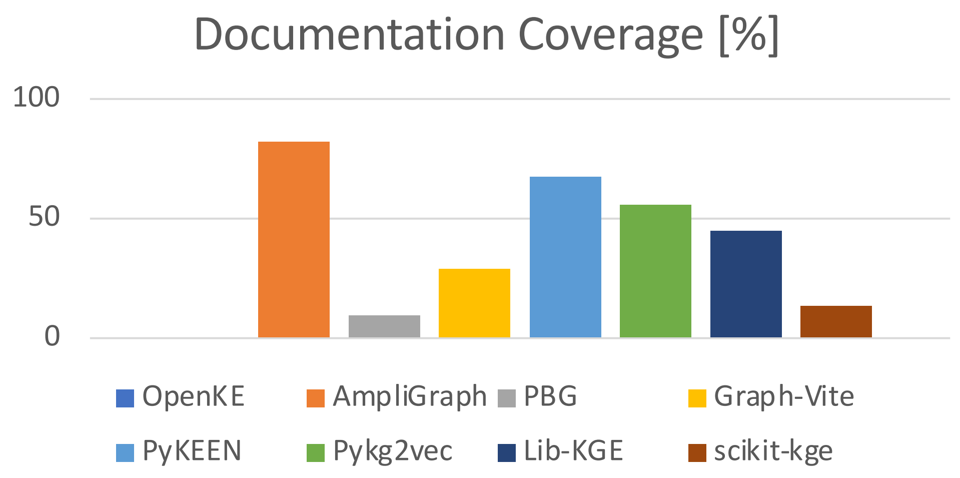
Formula: 10.0 - ((float(5 * error + warning + refactor + convention) / statement) * 10)

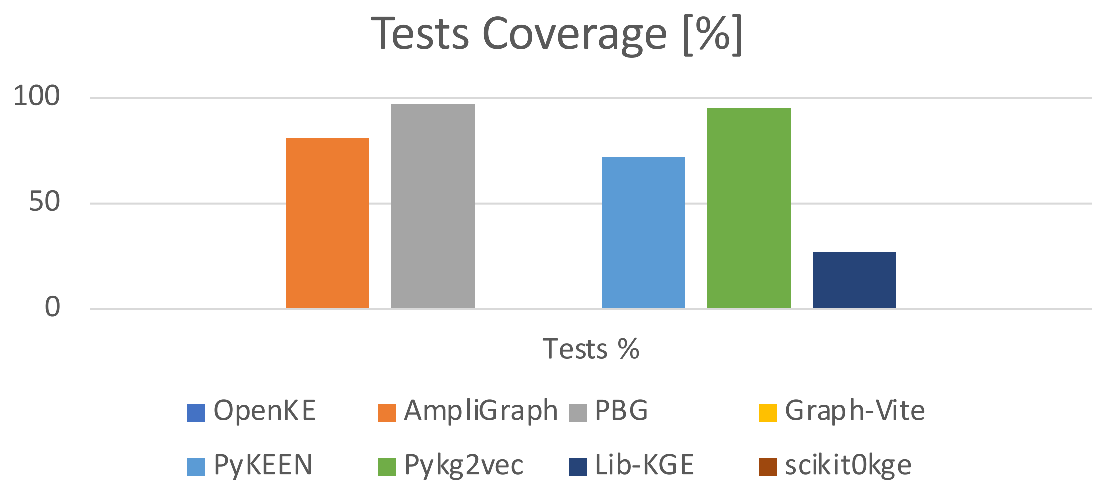
- Code Complexity (radon) – McCabe Complexity

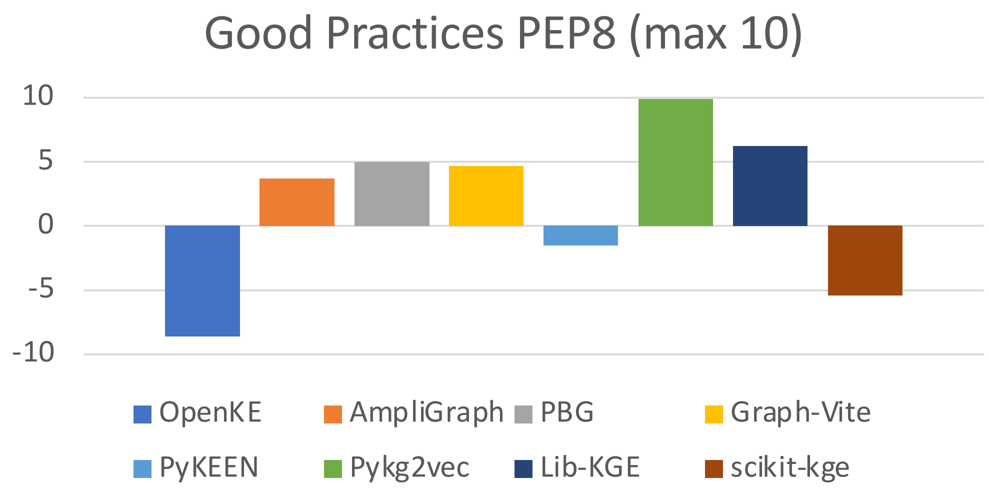
Class	A	B	C	D
Number	1-10	10-20	20-40	40+
Code	Well written and structured	Complex Code	Very Complex Code	Extremely Complex Code
Testability	High	Medium	Low	Not Testable
Maintenance Cost and Effort	Less	Medium	High	Very High

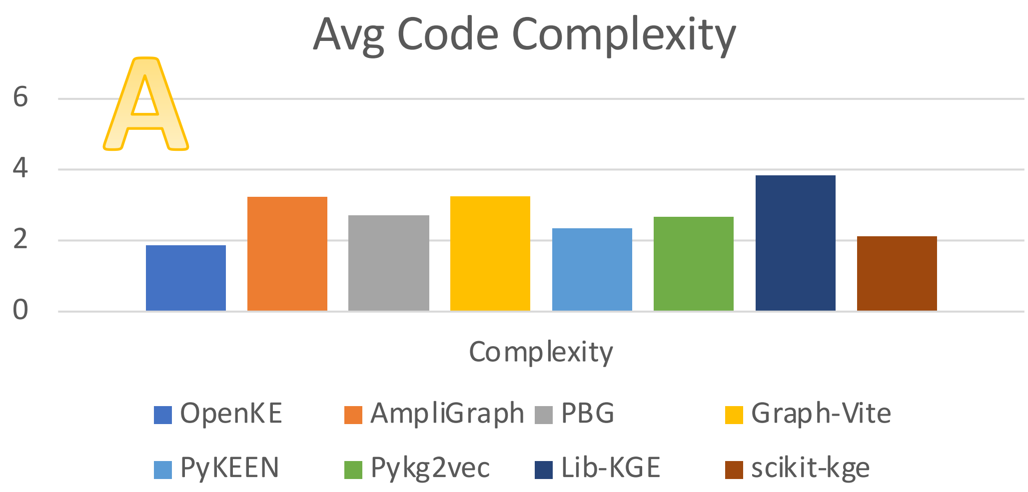
source

Software Development Metrics









Which Library Should I Use?

Align your choice with:

- Your task and time that you have for learning the library.
- Your experience.
- Framework the library supports Pytorch/Tensorflow/other?.
- Consider features like scalability, community support, user-friendliness, maturity of the project, accuracy, and supported addons.
- Finally: The choice is yours.

Use tools like github-statistics to support yourself.

Thank you!

Knowledge Graph Embeddings: From Theory to Practice

Software Ecosystem

adrianna.janik@accenture.com

Resources:

- AmpliGraph
- Libkge
- Graphvite
- DGL-KE
- Pykeen
- Pykg2vec
- OpenKE
- scikit-kge
- PyTorch-BigGraph
- github-statistics
- Article on how to compare repos

References

1. AmpliGraph: a Library for Representation Learning on Knowledge Graphs, Luca Costabello and Sumit Pai and Chan Le Van and Rory McGrath and Nicholas McCarthy and Pedro Tabacof, 2019, https://doi.org/10.5281/zenodo.2595043
2. Ali, Mehdi, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. "PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings." *ArXiv:2007.14175 [Cs, Stat]*, July 30, 2020. http://arxiv.org/abs/2007.14175.
3. Han, Xu, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. "OpenKE: An Open Toolkit for Knowledge Embedding," n.d., 6.
4. Lerer, Adam, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. "PyTorch-BigGraph: A Large-Scale Graph Embedding System," n.d., 12.
5. Nickel, Maximilian, Lorenzo Rosasco, and Tomaso Poggio. "Holographic Embeddings of Knowledge Graphs." *ArXiv:1510.04935 [Cs, Stat]*, December 7, 2015. http://arxiv.org/abs/1510.04935.
6. Ruffinelli, Daniel, Samuel Broscheit, and Rainer Gemulla. "LibKGE You CAN Teach an Old Dog New Tricks! On Training Knowledge Graph Embeddings," 2019. https://openreview.net/forum?id=BkxSmlBFvr.
7. Yu, Shih Yuan, Sujit Rokka Chhetri, Arquimedes Canedo, Palash Goyal, and Mohammad Abdullah Al Faruque. "Pykg2vec: A Python Library for Knowledge Graph Embedding." *ArXiv:1906.04239 [Cs, Stat]*, June 4, 2019. http://arxiv.org/abs/1906.04239.
8. Zheng, Da, Xiang Song, Chao Ma, Zeyuan Tan, Zihao Ye, Jin Dong, Hao Xiong, Zheng Zhang, and George Karypis. "DGL-KE: Training Knowledge Graph Embeddings at Scale." *ArXiv:2004.08532 [Cs]* Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (April 18, 2020). http://arxiv.org/abs/2004.08532.
9. Zhu, Zhaocheng, Shizhen Xu, Meng Qu, and Jian Tang. "GraphVite: A High-Performance CPU-GPU Hybrid System for Node Embedding." *The World Wide Web Conference on - WWW '19*, 2019, 2494–2504. https://doi.org/10.1145/3308558.3313508.
10. Bianchi, Federico, Gaetano Rossiello, Luca Costabello, Matteo Palmonari, and Pasquale Minervini. "Knowledge Graph Embeddings and Explainable AI." *ArXiv:2004.14843 [Cs]*, April 30, 2020. https://doi.org/10.3233/SSW200011.

Outline

Theoretical Overview — 1h 30m

- Introduction
- Anatomy of a Knowledge Graph Embedding Model
- Evaluation Protocol and Metrics
- Advanced KGE Topics
- Open Research Questions

Applications — 15 m

Software Ecosystem — 15 m

Hands-on Session — 1h 15m

bit.ly/kge-tutorial

Slides & Colab: kge-tutorial-ecai2020.github.io

♫ by La Fascination

Knowledge Graph Embeddings: From Theory to Practice

Luca Costabello
Accenture Labs
@lukostaz

Sumit Pai
Accenture Labs
@sumitppai

Nicholas McCarthy
Accenture Labs
@nickpmcc

Adrianna Janik
Accenture Labs
@adri_janik