Big Data Analytics Summer School, June 2020

Knowledge Graph Embedding

Dr. Sahar Vahdati

Department of
Computer
Science

The Challenges of Knowledge Graphs (KGs)

Knowledge Graph Creation Methods

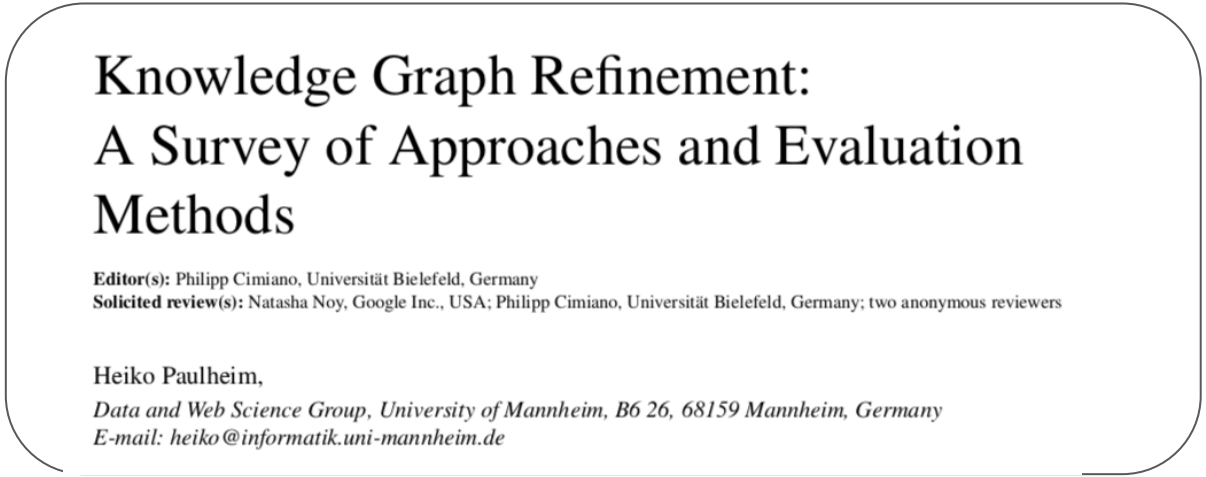
The represented knowledge can be **incorrect**.

The represented knowledge can be **incomplete**.

Knowledge Graph Refinement

ML-based Approaches for **Completion** of Knowledge Graphs

Error detection Approaches for **Correction** of Knowledge Graphs



Knowledge Graph Refinement:
A Survey of Approaches and Evaluation Methods

Editor(s): Philipp Cimiano, Universität Bielefeld, Germany
Solicited review(s): Natasha Noy, Google Inc., USA; Philipp Cimiano, Universität Bielefeld, Germany; two anonymous reviewers

Heiko Paulheim,
Data and Web Science Group, University of Mannheim, B6 26, 68159 Mannheim, Germany
E-mail: heiko@informatik.uni-mannheim.de

Graph Completion with Machine Learning

Hypothesis: ML can help to learn new knowledge from already existing ones!

Machine learning builds a **mathematical model** based on sample data

(set of attributes as training data D_n with n samples).

Supervised Learning

$$D_n = \{x_i, y_i\}_{i=1}^{n}$$

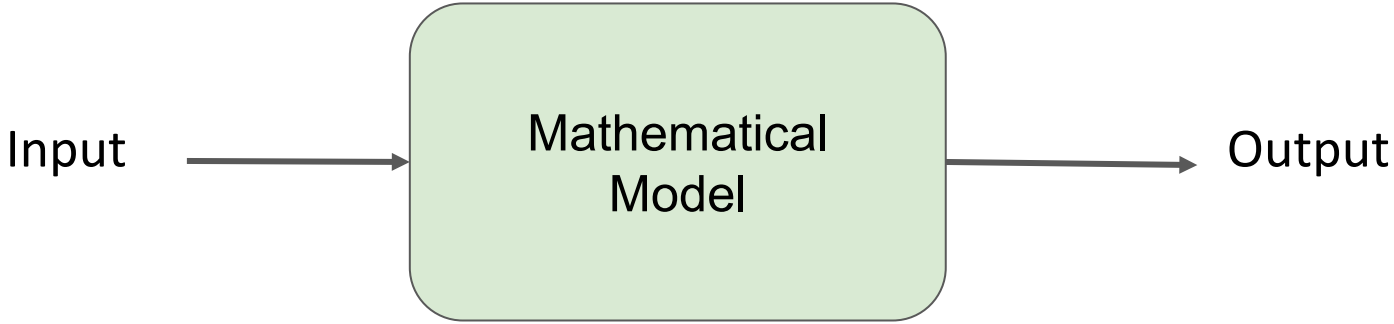
Un-Supervised Learning

$$D_n = \{x_i\}_{i=1}^{n}$$

Semi-Supervised Learning

$$D_n = \{x_i, y_i\}_{i=1}^{n_1} \cup \{x_i\}_{i=n_1+1}^{n}$$

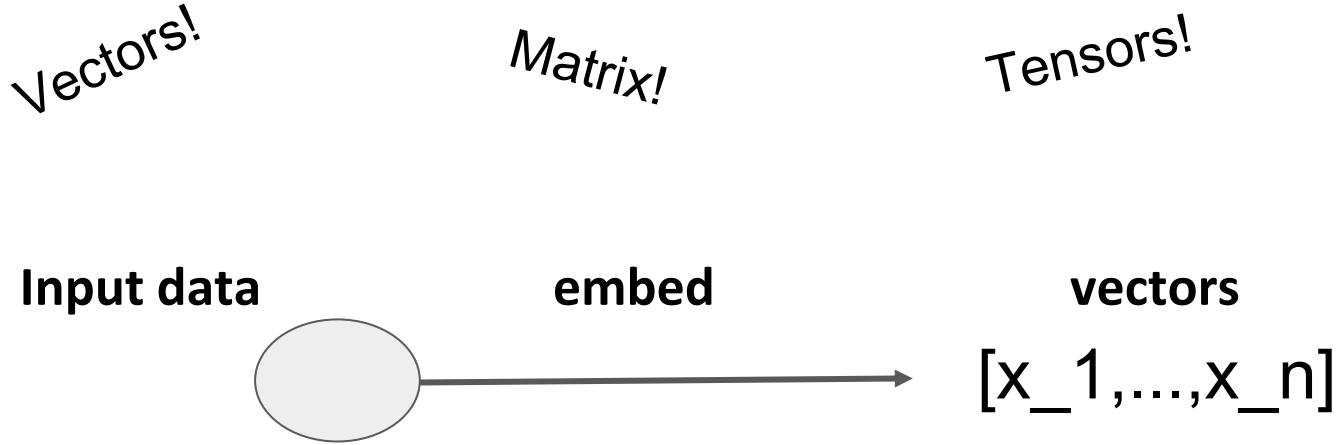
Ml-based Mathematical Model for Learning



$$f(X) = Y$$

Requirement

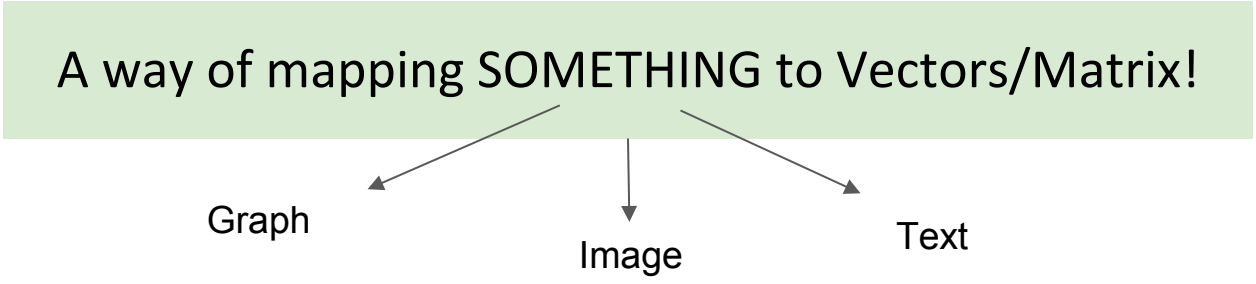
The input and output of such a Mathematical Model should be numeric!



What is an embedding?

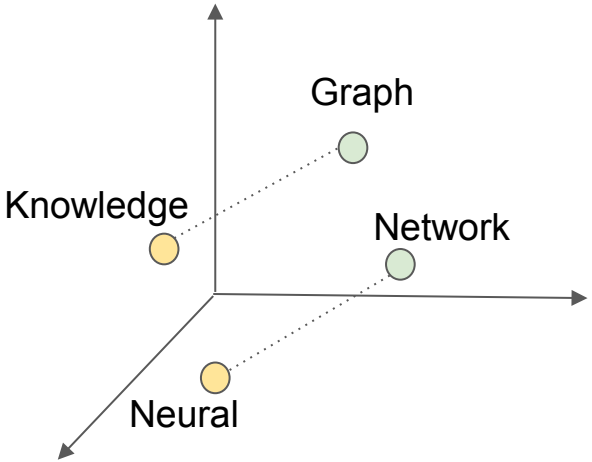
Wikipedia: In mathematics, an embedding is one instance of some mathematical structure contained within another instance, such as a group that is a subgroup.

Google: An embedding is a relatively low-dimensional space into which you can translate high-dimensional vectors.

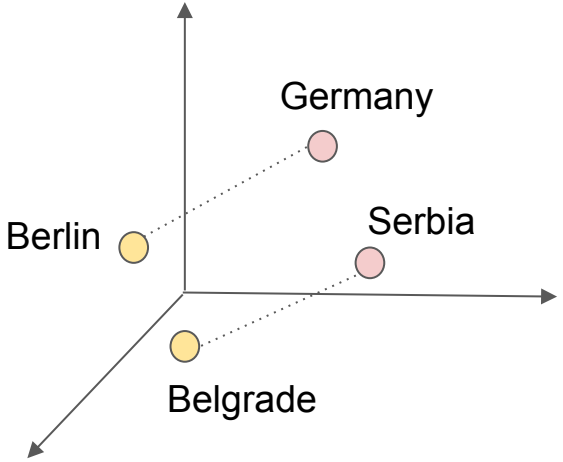


Word Embedding

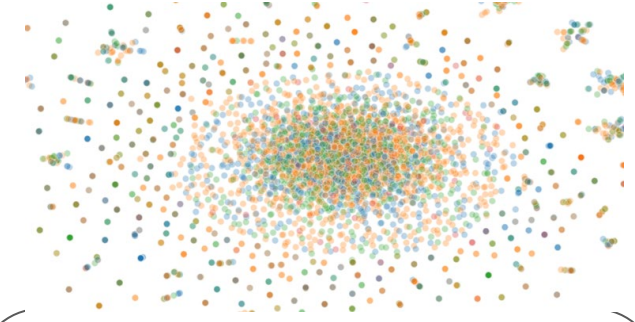
Knowledge [1.2,1.3,2.0]
Graph [1.4,1.0,1.9]
Neural [3.3,4.3,2.5]
Network [3.1,2.9,2.4]

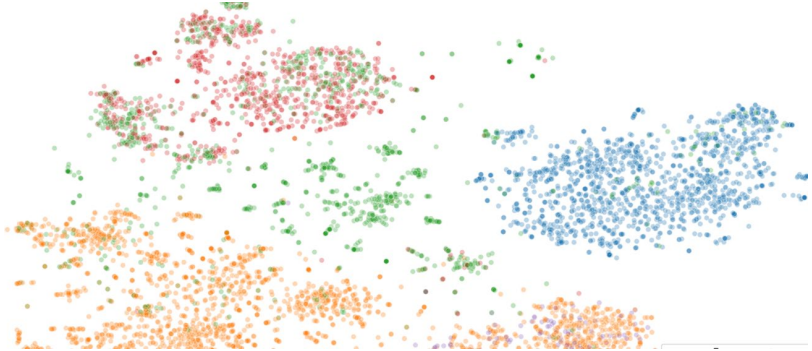


Berlin [1.2,1.3,2.0]
Germany [1.4,1.0,1.9]
Belgrade [3.3,4.3,2.5]
Serbia [3.1,2.9,2.4]



Graph Embedding

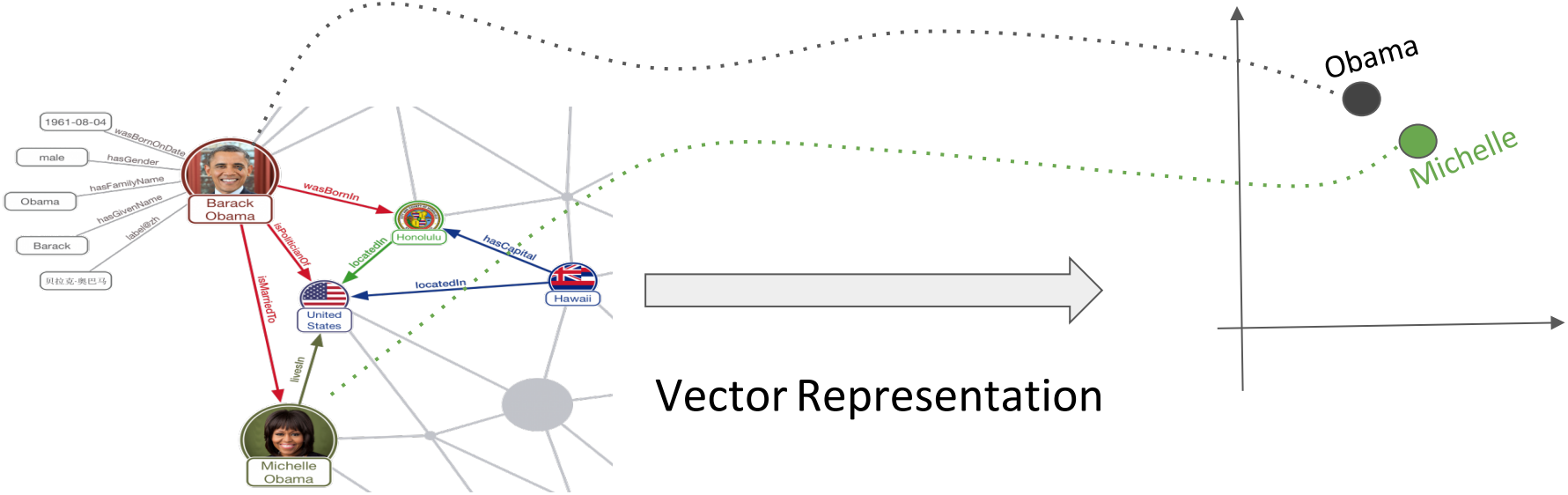
f () =



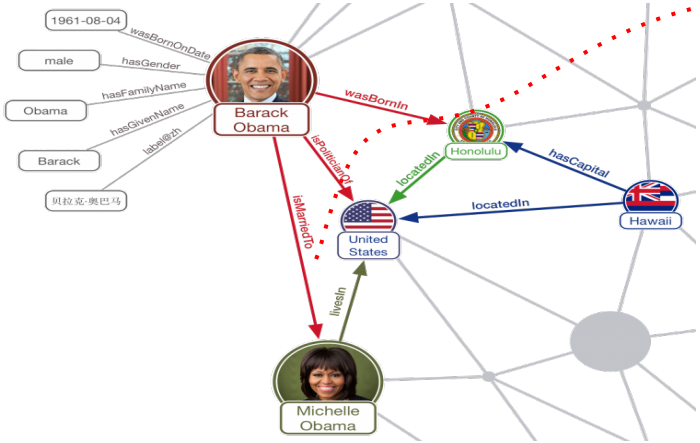
0000001	1000110	0101010	0011100	1101101	0110111	1011011	1110000
0000010	1000101	0101001	0011111	1101110	0110100	1011000	1110011
0000100	1000011	0101111	0011001	1101000	0110010	1011110	1110101
0001000	1001111	0100011	0010101	1100100	0111110	1010010	1111001
0010000	1010111	0111011	0001101	1111100	0111110	1001010	1100001
0100000	1100111	0001011	0111101	1001100	0010110	1111010	1010001
1000000	0000111	1101011	1011101	0101100	1110110	0011010	0110001
0000011	1000100	0101000	0011110	1101111	0110101	1011001	1110010
0000110	1000001	0101101	0011011	1101010	0110000	1011100	1110111
0001100	1001011	0100111	0010001	1100000	0111010	1010110	1111101
0011000	1011111	0110011	0000101	1110100	0101110	1000010	1101001
0001010	1001101	0100001	0010111	1100110	0111100	1010000	1111011
0010100	1010011	0111111	0001001	1111000	0100010	1001110	1100101
0010010	1010101	0111001	0001111	1111110	0100100	1001000	1100011
0001110	1001001	0100101	0010011	1100010	0111000	1010100	1111111

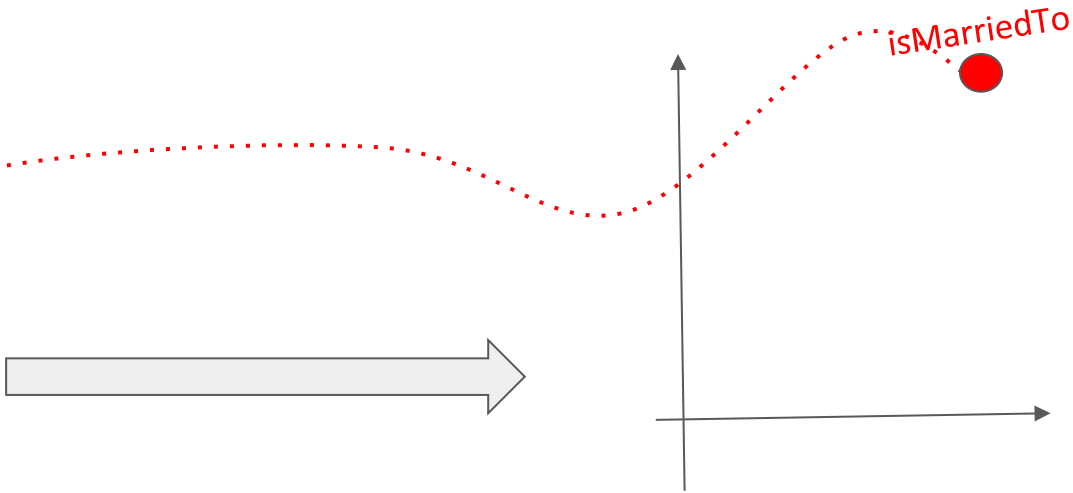
$$\begin{pmatrix} 0.55 & 0.66 \\ \vdots & \\ 0.75 & 0.16 \end{pmatrix}$$

Node Embedding



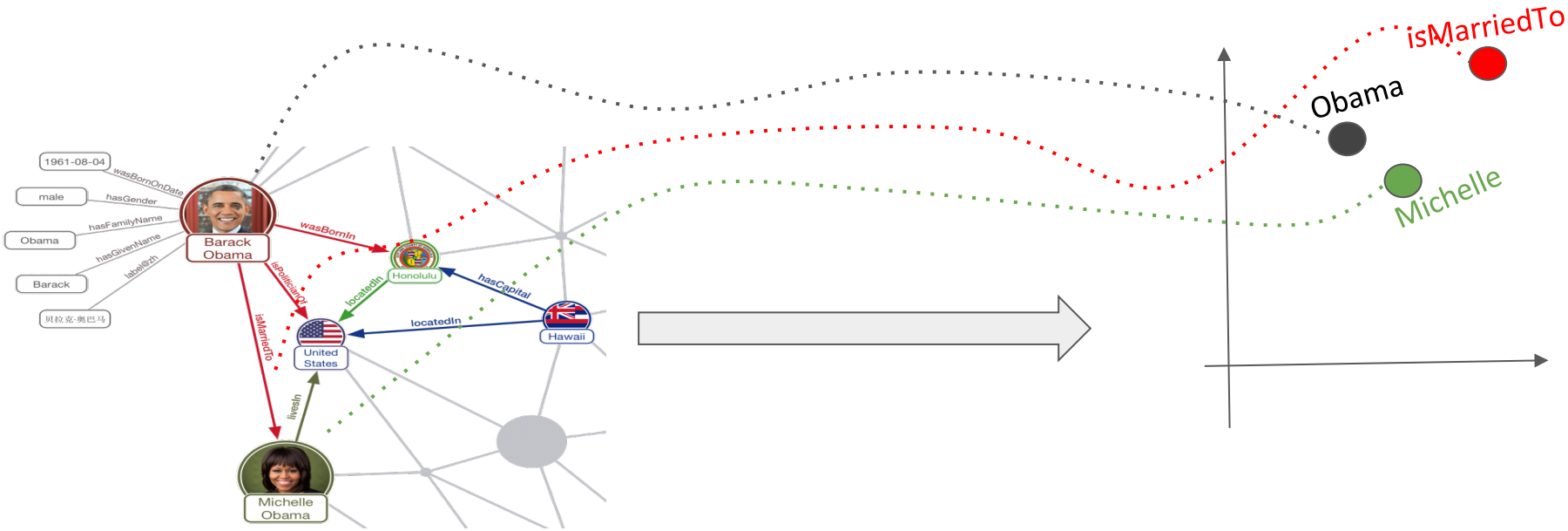
Edge Embedding





Vector Representation

Knowledge Graph Embedding

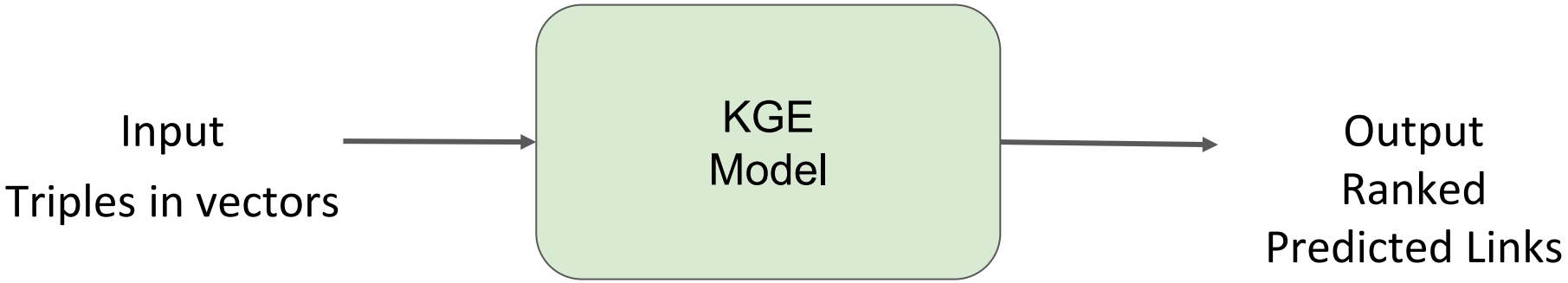


Symbolic Representation

Vector Representation

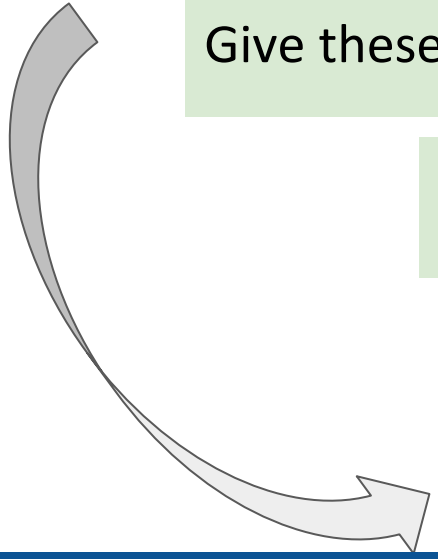
Knowledge Graph Embedding Models

Goal: Predicting missing links between nodes!



Learning Steps of a KGE

Assign random vectors to each entity instance or relation!



Give these vectors to the KGE model

Let the model learn the embeddings

Return degree of plausibility

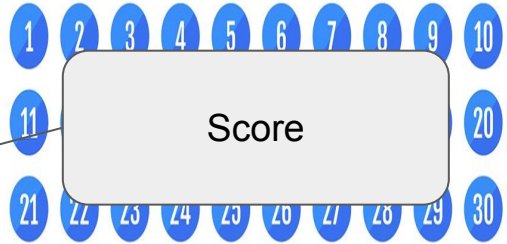
Optimize the learned embeddings

Designing Knowledge Graph Embeddings

Data Sets/KGs

KGE

Score



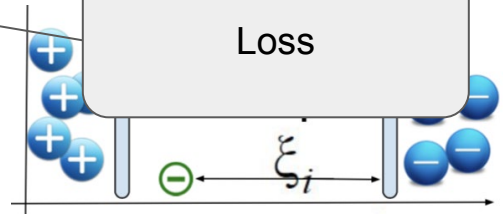
Embeddings

[0.1, -0.4, 1, ...]

Vectors, Matrix, ...

2	5	7	8
1	2	3	1
4	5	0	1

Loss

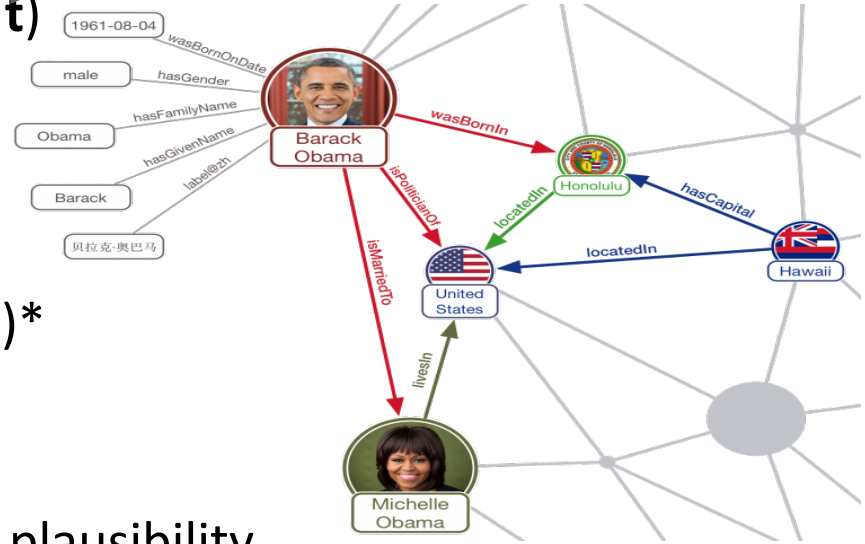


Datasets/KGs

Dataset: Knowledge Graph

Triple facts are shown in the form of (h,r,t)

Vectors are shown in the form of $(\vec{\mathbf{h}}, \vec{\mathbf{r}}, \vec{\mathbf{t}})$



(Barack Obama, WasBornIn, Honolulu)

$(\overrightarrow{\textbf{Barack Obama}}, \overrightarrow{\textbf{WasBornIn}}, \overrightarrow{\textbf{Honolulu}})^*$

$x\_i = (\vec{\mathbf{h}}, \vec{\mathbf{r}}, \vec{\mathbf{t}})^*$

y_i= 1 (True), 0(False) with a degree of plausibility.

Learning Datasets

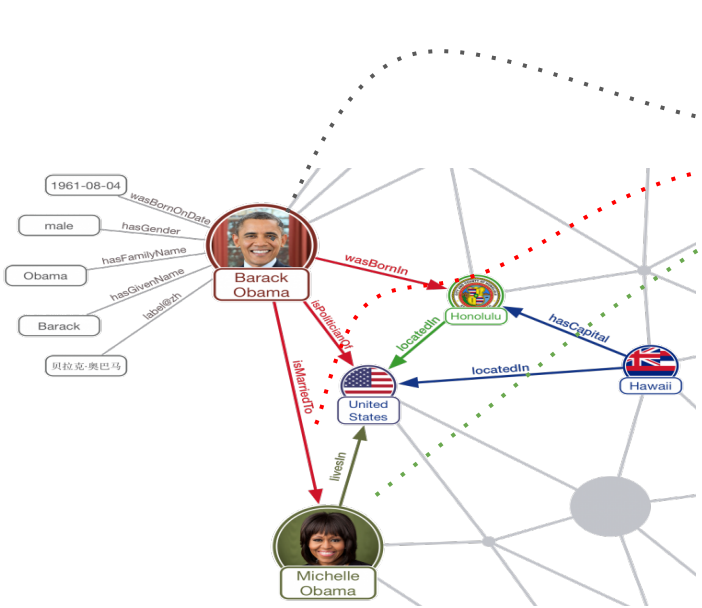
Train: Use to learn embeddings.

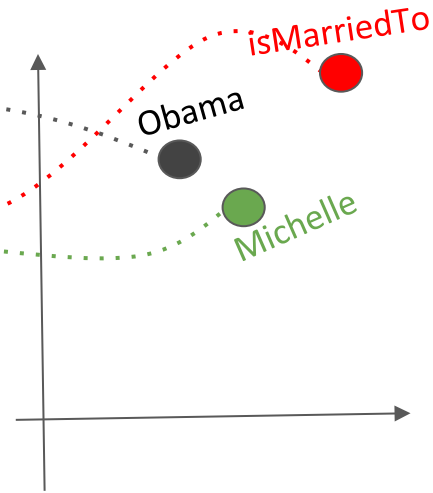
Test: Check the correctness of the results on one gold standard.

Validation: Check the behavior of the model in different situations.

Negative Sampling: let the model also learn from incorrect samples.

How to arrange the vectors?





Symbolic Representation

Vector Representation

Score function

Provides the degree of correctness for a triple:

TransE Score Function

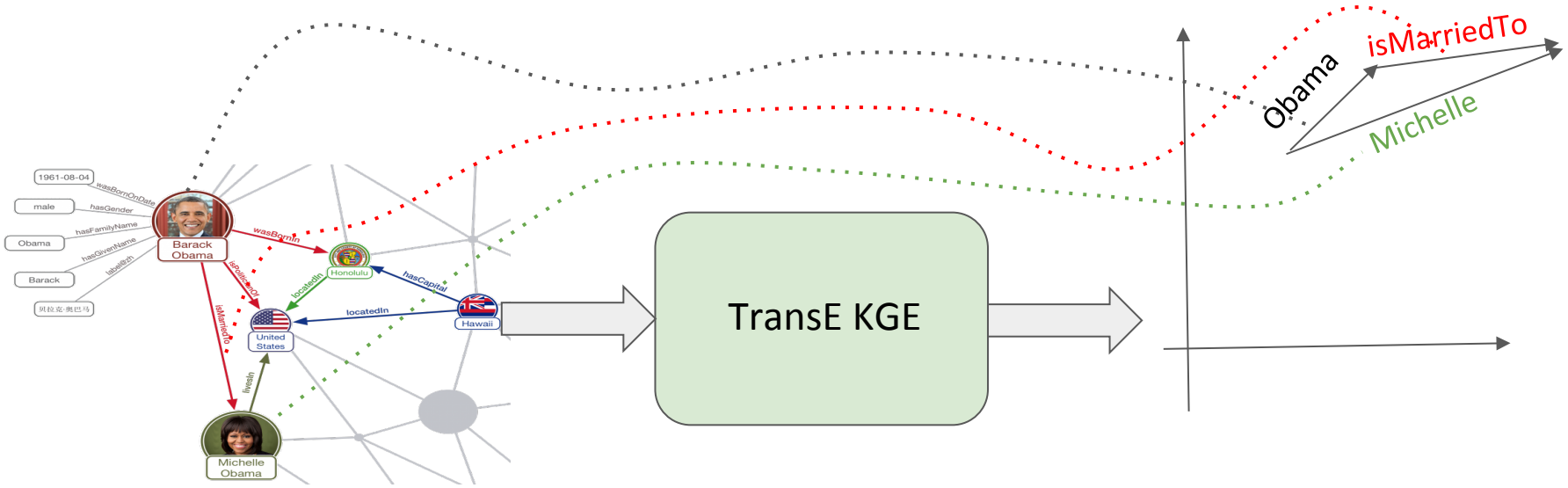
$$f_r^{TransE}(h,t) = \|\overrightarrow{h} + \overrightarrow{r} - \overrightarrow{t}\|$$

$$f_{wasBornIn}(\overrightarrow{BarackObama}, \overrightarrow{Honolulu}) = 0, \rightarrow (BarackObama, wasBornIn, Honolulu)\ is\ True$$

$$f_{wasBornIn}(\overrightarrow{BarackObama}, \overrightarrow{Berlin}) = 9 \rightarrow (BarackObama, wasBornIn, Berlin)\ is\ False$$

Purpose: Vector Arrangements

TransE Embedding Model: $\overrightarrow{\text{subject}} + \overrightarrow{\text{property}} = \overrightarrow{\text{object}}$

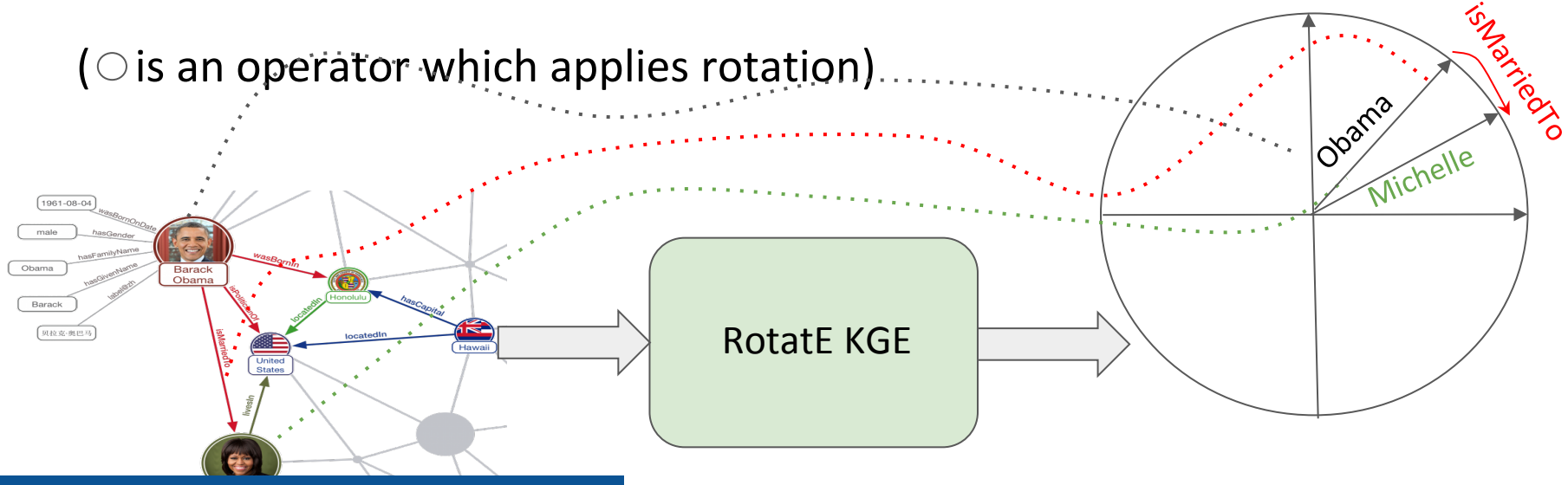


Purpose: Vector Arrangements

RotatE Score Function

$\overrightarrow{subject} \circ \overrightarrow{property} = \overrightarrow{object}$

($\circ$ is an operator which applies rotation)



Optimization

To optimize the random vectors in a way that the scores are closer to what we have in the KG and the model knows about them!

In order to design an optimizer, we need to know what do we want to optimize with a criteria.

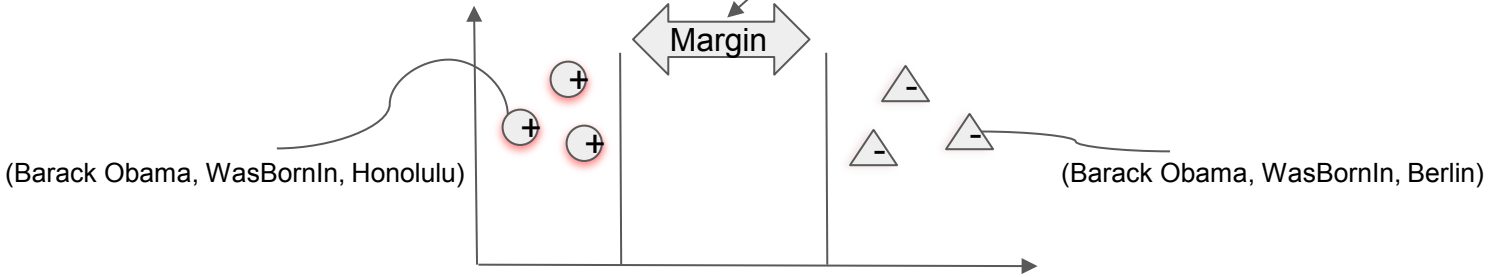
Margin: The difference between correctness of positive triples and negative one

Loss Function

Adjust the embedding vectors for entities and relations i.e., (h, r, t) to enforce the criteria.

Margin Ranking Loss:

$$Loss = \|\lambda + f_r(h,t) - f_r(h^{'},t^{'})\|_+$$

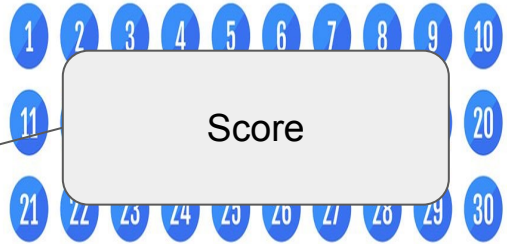


Designing Knowledge Graph Embeddings

Data Sets/KGs

KGE

Score



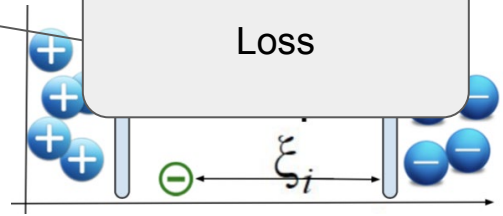
Embeddings

[0.1, -0.4, 1, ...]

Vectors, Matrix, ...

2	5	7	8
1	2	3	1
4	5	0	1

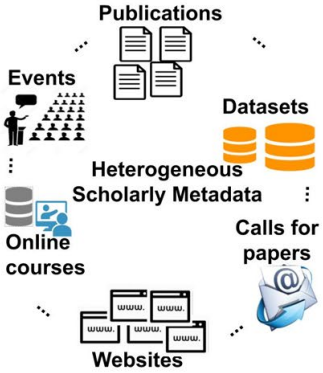
Loss

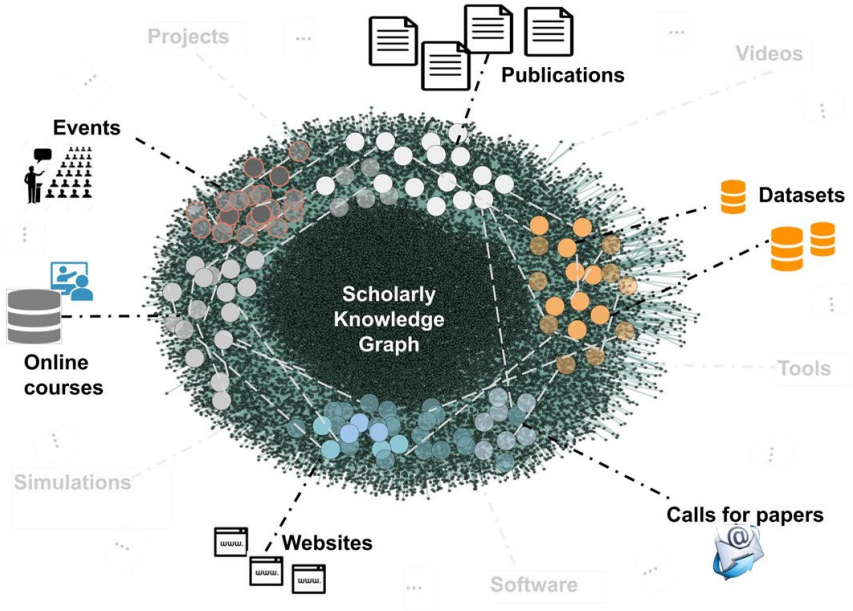


Example use-case
Co-author Recommendation

Heterogeneous Scholarly Metadata and Enquiries!

- Enormous resources / metadata providers
- Different types
- Different formats
- Large-scale metadata
- Diverse
- Broad scientific domains
- Value and importance
- Increasing publishing rate





Machine Learning Support for Scholarly Domain

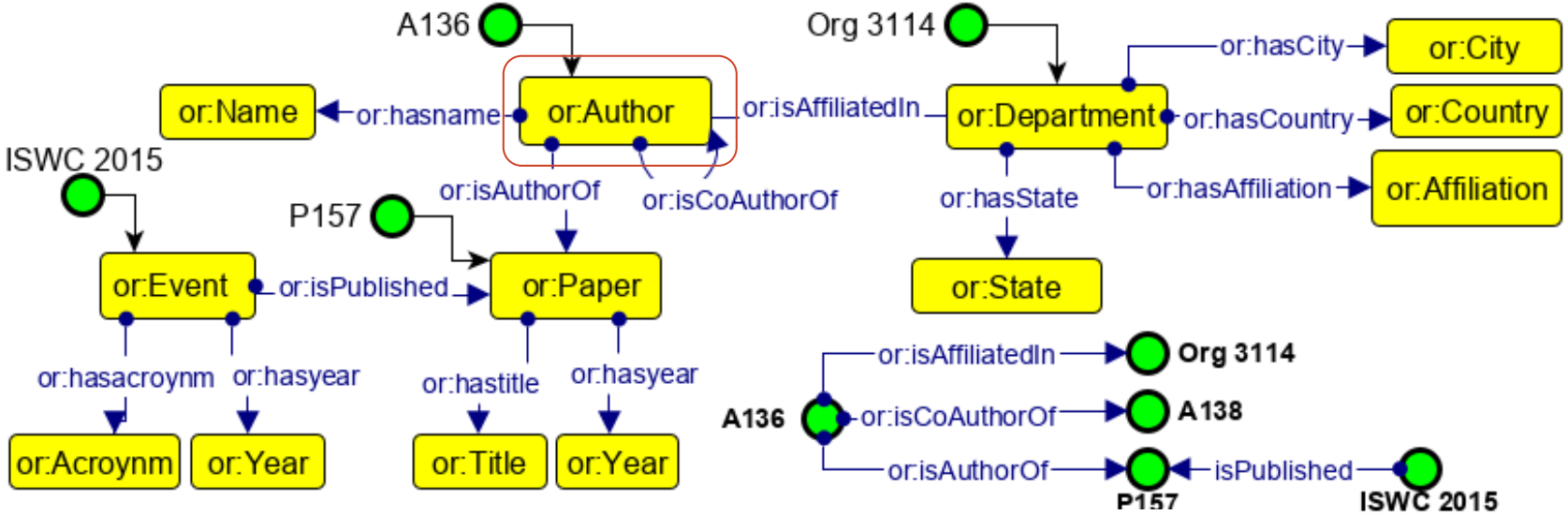
Many metaresearch enquiries of scholars remain unrevealed.

Link prediction for recommendation-based services:

- Who can be the best candidate for collaboration?
- Which group/university can be the best candidate for collaboration?
- Which groups in different fields can work on an effective research together?
- …

ML-based approaches for link prediction: **Knowledge Graphs Embedding**

Ontology of an Scholarly Knowledge Graph

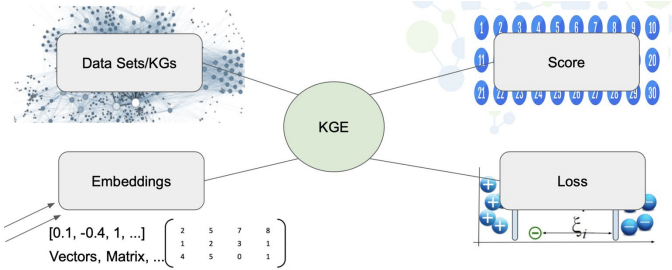


	Entities			Relations		
Dataset	Author	Publication	Venue	hasAuthor	hasCoauthor	hasVenue
SKGOLD	4,660	2,870	7	9,934	12,921	6,614
SKGNEW	12,472	5,001	42	14,933	21,279	5,001

Knowledge Graphs Embedding Models

TransE: $f_r(h,t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$

RotatE: $f_r(h,t) = \|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|$



Loss Functions

$$\mathcal{L} = \sum_{(h,r,t)\in S^+} \sum_{(h',r',t')\in S^-} [f_r(h,t) + \gamma - f_r(h',t')]_+$$

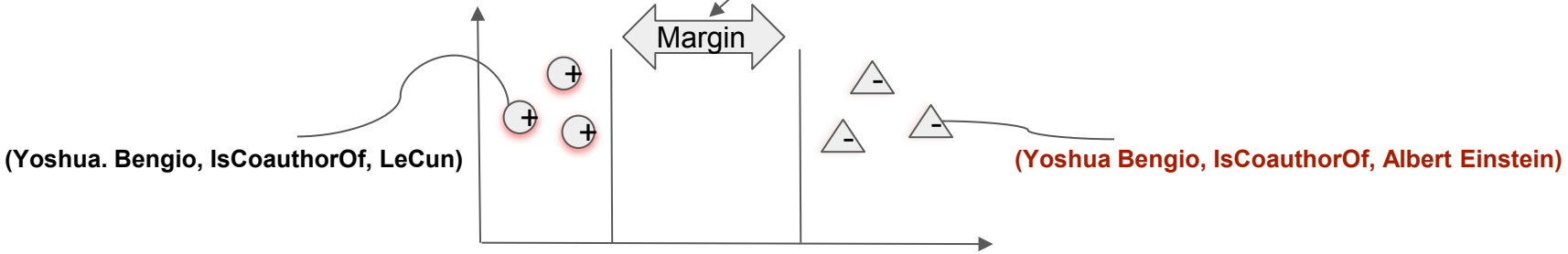
$$\mathcal{L}_{RS} = \sum\sum [f_r(h,t) + \gamma - f_r(h',t')]_+ + \lambda[f_r(h,t) - \gamma_1]_+$$

Loss function

Loss function helps to obtain embedding vectors for entities and relations i.e., (h, r, t).

Margin Ranking Loss:

$$Loss = \|\gamma + f_r(h,t) - f_r(h^{'},t^{'})\|_+$$



Loss function

Problems of Margin Ranking loss:

- Margin sliding: there are infinite solutions for the score of positive and negative samples

$$f_r(h,t) = 0 \text{ and } f_r(h',t') = \gamma, \text{ or}$$
$$f_r(h,t) = \gamma \text{ and } f_r(h',t') = 2\gamma, \text{ or}$$
$$\vdots$$
$$f_r(h,t) = (n-1)\gamma \text{ and } f_r(h',t') = n\gamma.$$

- Embeddings are adversely affected by false negative samples.

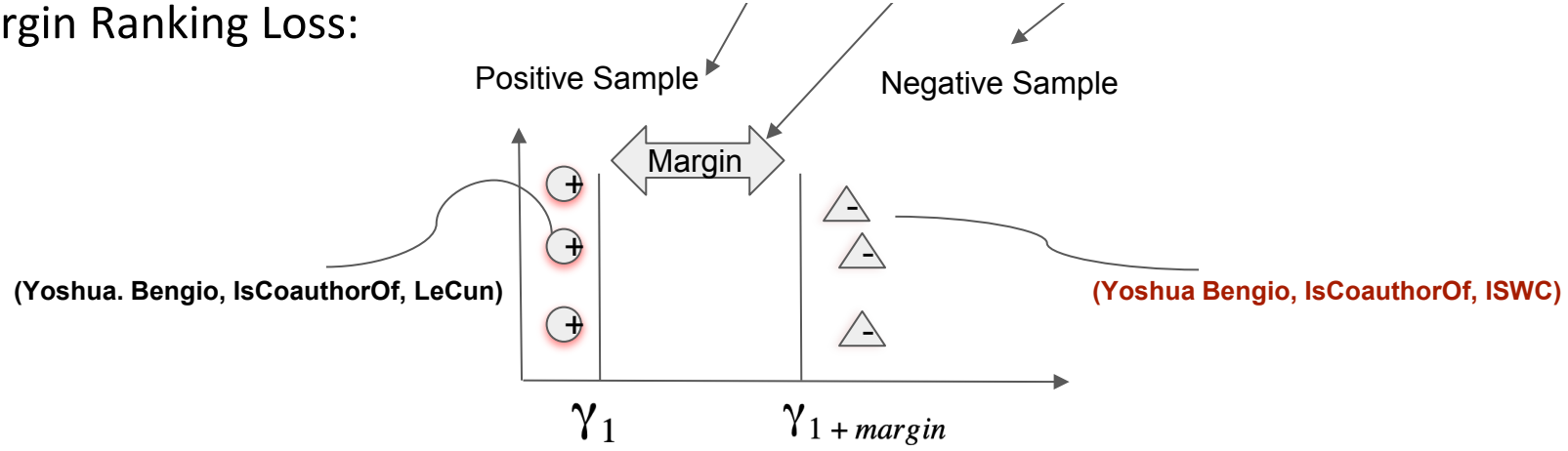
Loss function

Limit-based scoring loss:

Upper bound of positive sample

$$\mathcal{L}_{RS} = \sum\sum [f_r(h,t) + \gamma - f_r(h',t')]_+ + \lambda[f_r(h,t) - \gamma_1]_+$$

Margin Ranking Loss:



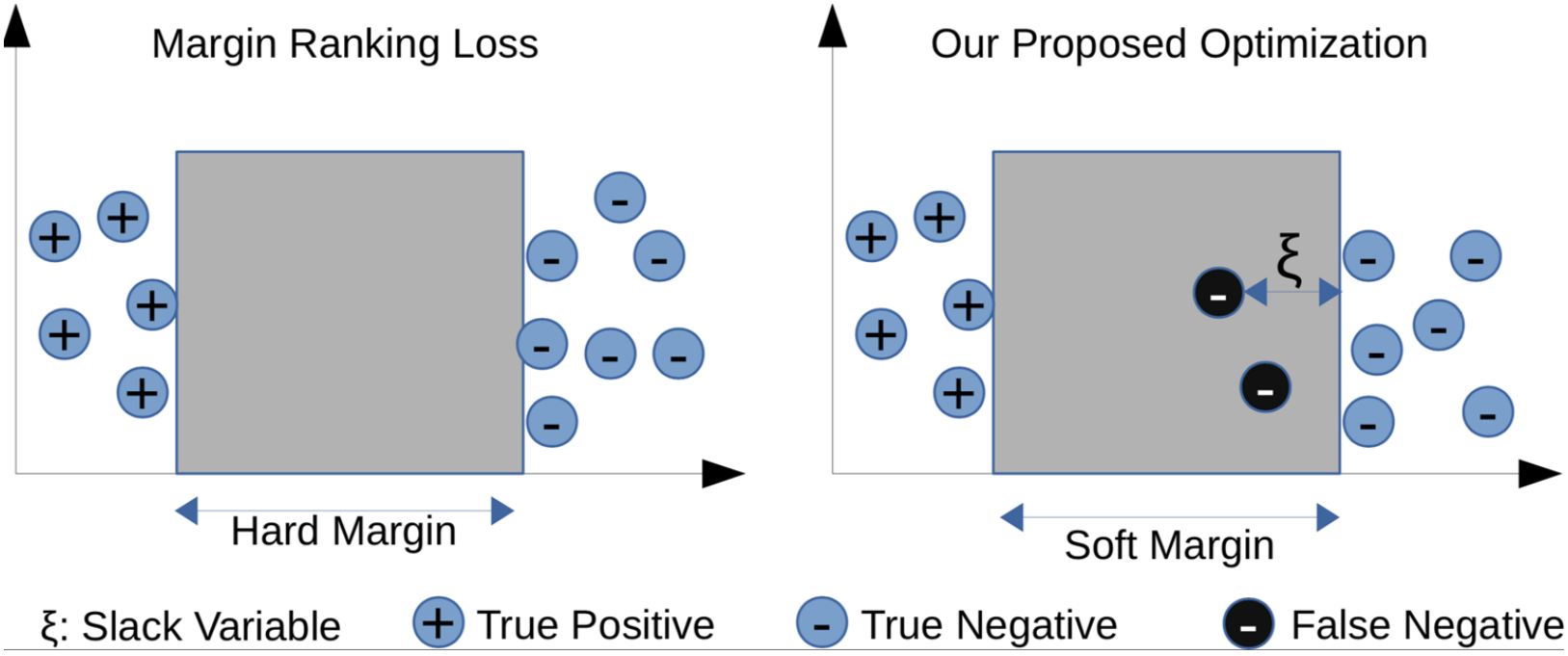
Challenge of Co-author Recommendation

Problems:

- Many to Many relation
  - Coauthor relation
  - Citations
- In the existence of many-to many relations, the rate of false negative samples increases.

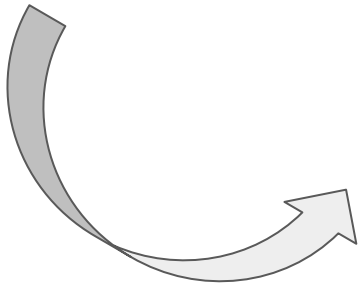
- Generating negative samples are based on a random corruption.
- Assuming that N= 1000 is the number of all authors in a SKG, the probability of generating false negatives for an author with 100 true or sensible but unknown collaborations becomes 100/1000= 10%.

Optimization of Margin ranking loss



Soft Marginal Loss

$$
\begin{cases}
\min_{\xi^r_{h,t}} \sum_{(h,r,t)\in S^+} {\xi^r_{h,t}}^2 \\
\text{s.t.} \\
f_r(h,t) \le \gamma_1, \ (h,r,t) \in S^+ \\
f_r(h',t') \ge \gamma_2 - \xi^r_{h,t}, \ (h',r,t') \in S^- \\
\xi^r_{h,t} \ge 0
\end{cases}
$$



$$
\min_{\xi^r_{h,t}} \sum_{(h,r,t)\in S^+} (\lambda_0 {\xi^r_{h,t}}^2 + \lambda_1 \max(f_r(h,t) - \gamma_1, 0) + \sum_{(h',r,t')\in S^-_{h,r,t}} \lambda_2 \max(\gamma_2 - f_r(h',t') - \xi^r_{h,t}, 0))
$$

Experiments and Results

Filtered mean rank

Filtered hits@1o

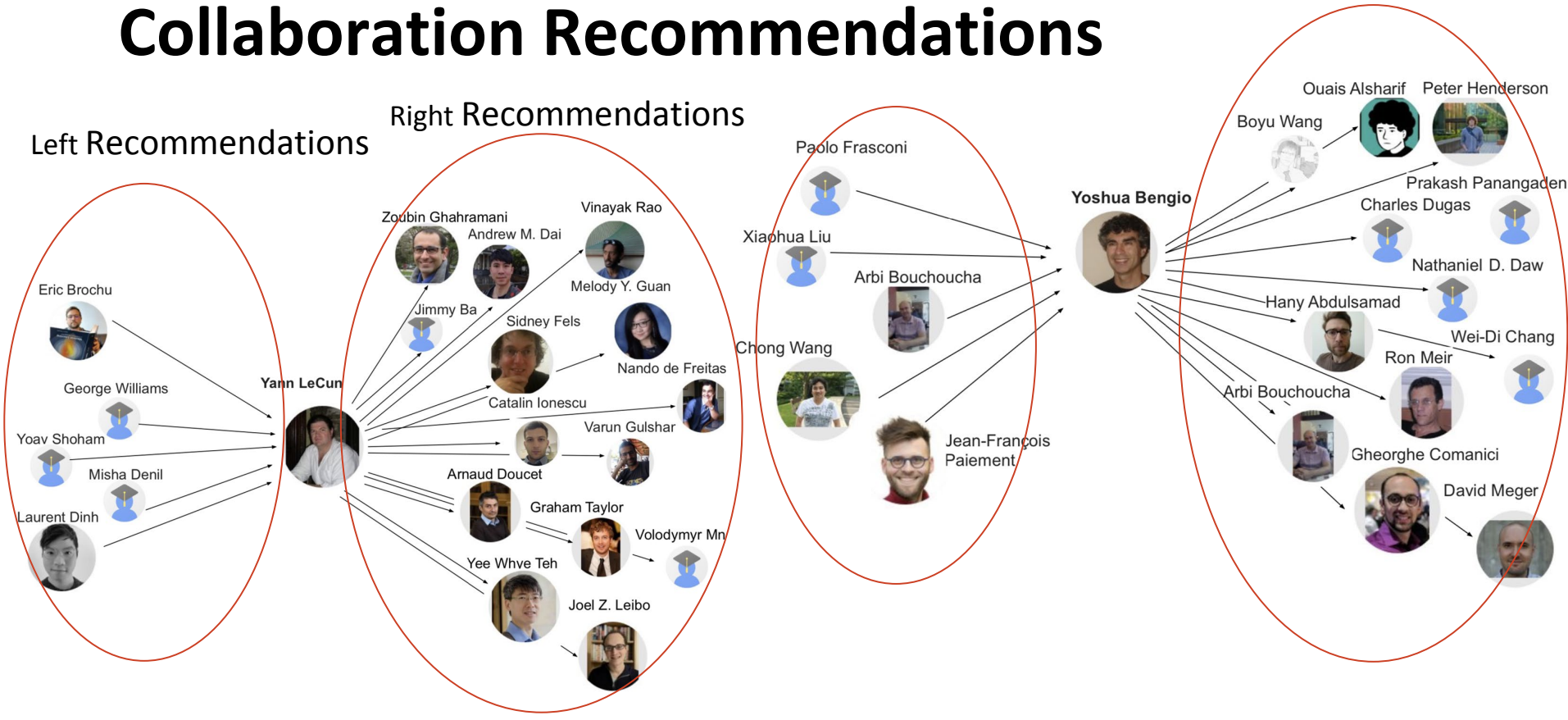
Filtered mean reciprocal rank

	SKGOLD – Filtered			SKGNEW – Filtered		
	FMR	FHits@10	FMRR	FMR	FHits@10	FMRR
TransE [4]	<u>647</u>	50.7	-	1150	<u>77.2</u>	-
ComplEx [19]	-	56.2	0.326	-	73.9	<u>0.499</u>
ConvE [7]	1215	49.3	0.282	1893	71.3	0.442
RotatE [15]	993	<u>60.6</u>	<u>0.346</u>	1780	69.5	0.486
TransE-RS [28]	-	-	-	<u>762</u>	75.8	0.443
TransE-SM (our work)	910	**61.4**	**0.347**	**550**	**79.5**	0.430
RotatE-SM (our work)	990	60.9	0.347	1713	76.7	**0.522**

Experiments and Results

	FB15k			WN18		
	FMR	FMRR	FHits@10	FMR	FMRR	FHits@10
TransE [4]	125	-	47.1	251	-	89.2
ComplEx [19]	106	67.5	82.6	543	94.1	94.7
ConvE [7]	51	68.9	85.1	504	94.2	95.5
RotatE [15]	49	68.8	85.9	388	94.6	95.5
TransE-RS [28]	38	57.2	82.8	189	47.9	95.1
TransE-SM	46	64.8	**87.2**	201	47.8	95.2
RotatE-SM	**40**	**70.4**	**87.2**	213	**94.7**	**96.1**

Collaboration Recommendations



More Application of KGEs

Question Answering Systems

Recommendation Systems

Prediction Systems

.

.

.

Workshop of Knowledge Representation & Representation Learning

ECAI 2020 in Santiago de Compostela, June 2020

Thank you!