

Knowledge Graph Embeddings

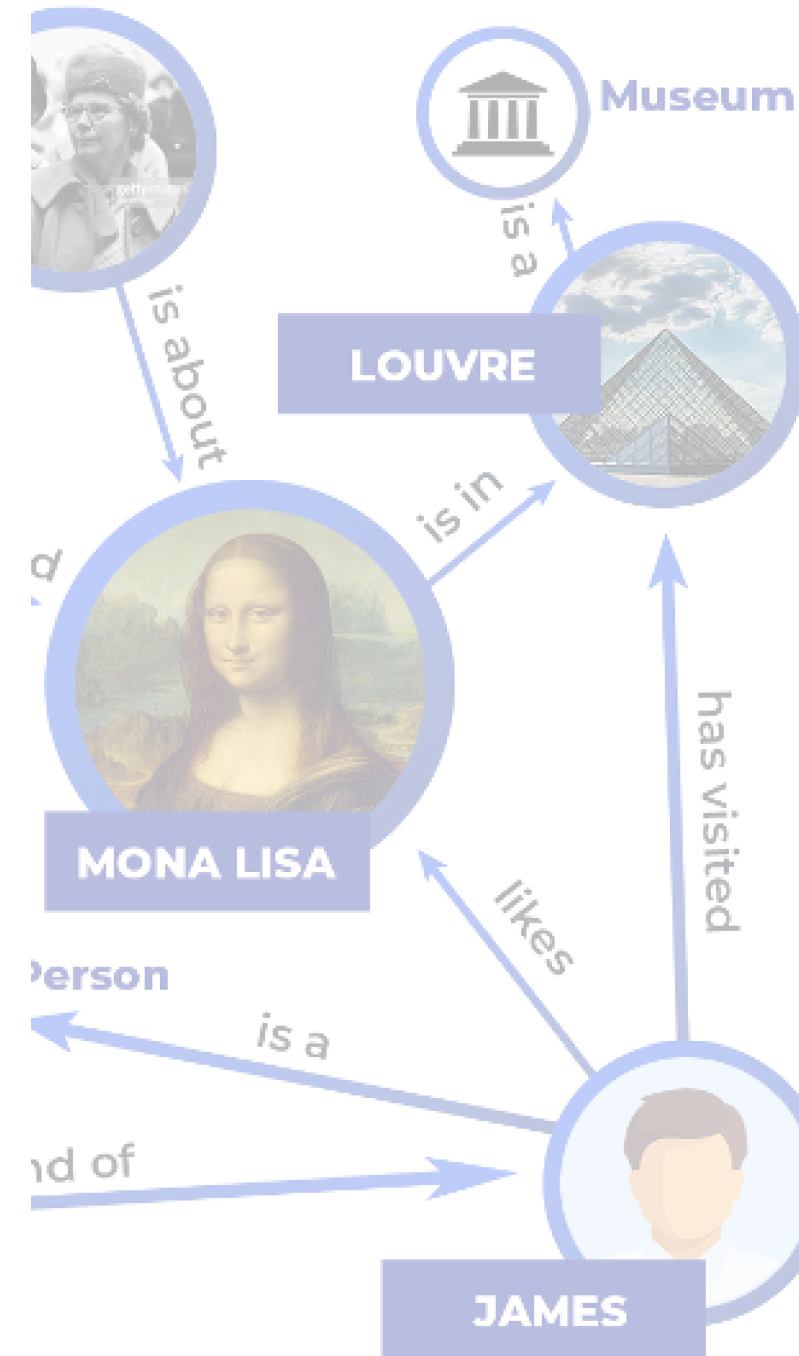
Apprendimento Automatico

A.A. 2024-2025

Docente: **Nicola Fanizzi**

Speaker: **Ivan Diliso**

Dipartimento di Informatica
Università Degli Studi di Bari Aldo Moro





Chi sono?

Ivan Diliso

Dottorando in Informatica e Matematica del gruppo di ricerca **ARA** (Apprendimento e Ragionamento Automatico).

Di cosa mi occupo?

Knowledge Graph Embedding, Neuro-Symbolic AI, Ontologie e Ontology Injection

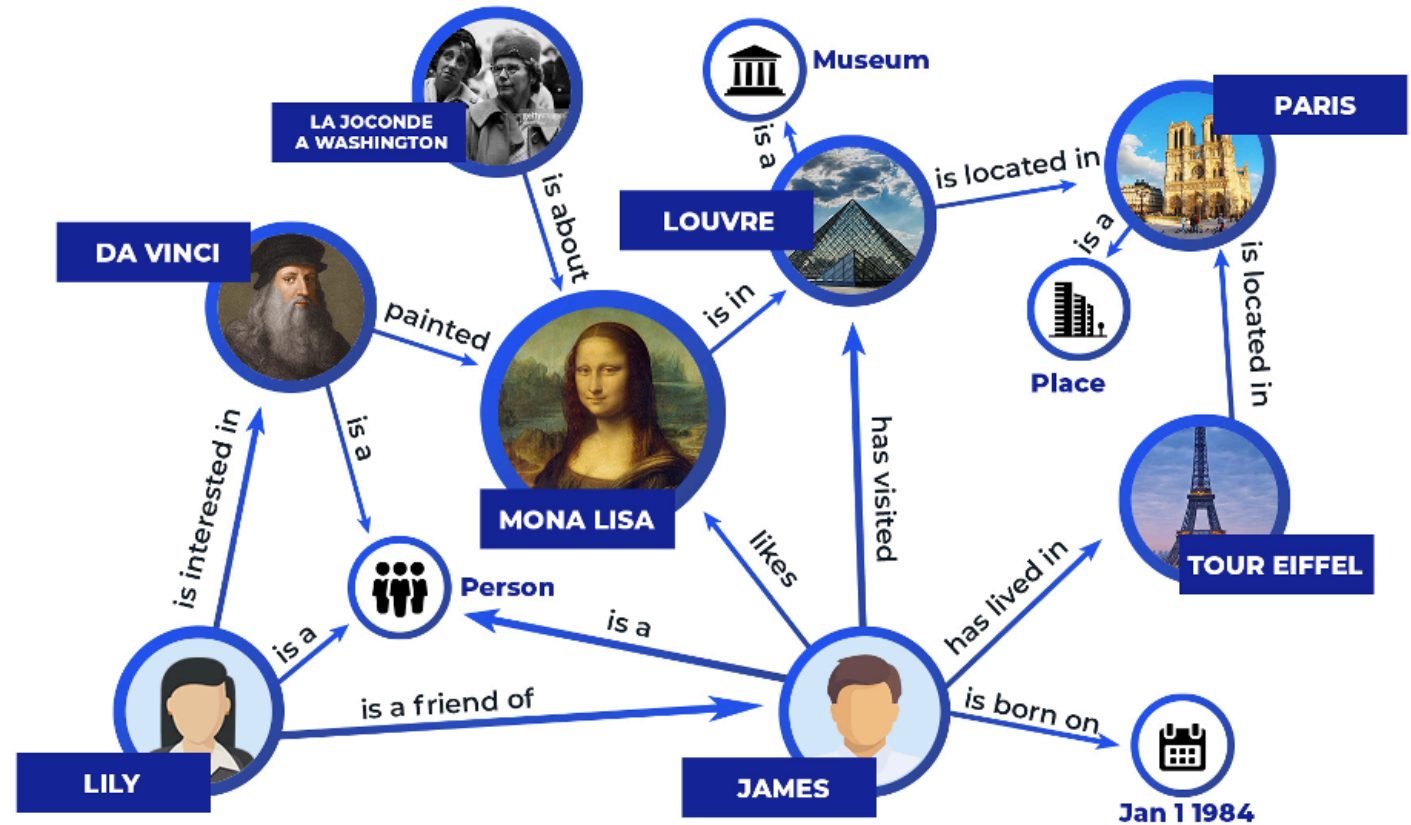
Indice dei contenuti

1. Cosa è un **Knowledge Graph**
2. **Machine Learning** su Knowledge Graphs
3. Strategie di **Valutazione** e Metriche
4. Argomenti Avanzati
 - Calibrazione
 - Ragionamento Neuro-Simbolico
 - Spazi di rappresentazione oltre l'euclideo

Definizioni

Fondamentali

Un **Knowledge Graph** è una rappresentazione strutturata di conoscenza tramite nodi (**entità**) e archi (**relazioni**) per collegare e integrare informazioni.

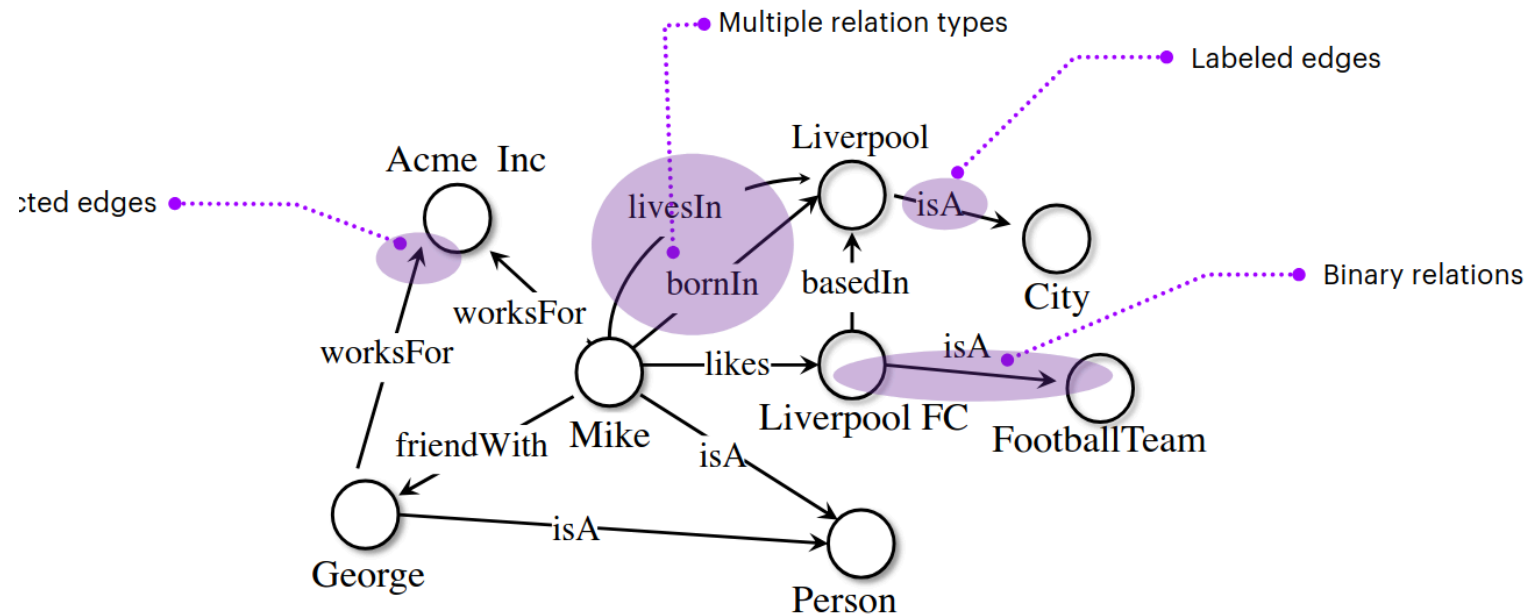


Definizioni

Fondamentali

Sui dati, possiamo fare due tipi di assunzioni fondamentali:

- **Closed** World Assumption (CWA): Assenza di un fatto, significa che è falso
- **Open** World Assumption (OWA): Assenza di un fatto, significa solo che non è conosciuto



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

\mathcal{E} : set of entities of \mathcal{G}

\mathcal{R} : set of relations of \mathcal{G}

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

Definizioni Fondamentali

Questo tipo di dati si presenta sottoforma di **triple**, nella forma $\langle s, p, o \rangle$, ad esempio, potremmo avere:

$$\langle \textit{Mario}, \textit{friendOf}, \textit{Luigi} \rangle$$

Quando parlo di **soggetto** della tripla, mi sto riferendo alla entità in testa alla tripla (eg. Mario), con **oggetto** mi riferisco invece alla entità in coda alla tripla (eg. Luigi)

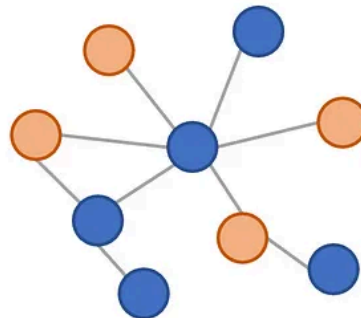
Machine Learning su Knowledge Graphs

Problemi risolvibili:

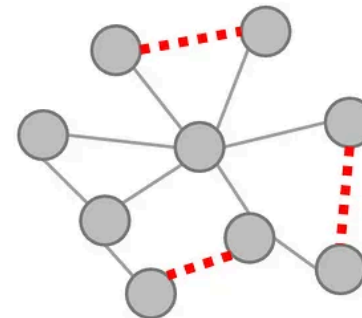
- Link Prediction
- Triple Classification
- Graph Classification
- Community Detection
- Anomaly Detection
- Entity Matching
- ...

Image from: <https://towardsdatascience.com/graph-neural-networks-with-pyg-on-node-classification-link-prediction-and-anomaly-detection-14aa38fe1275/>

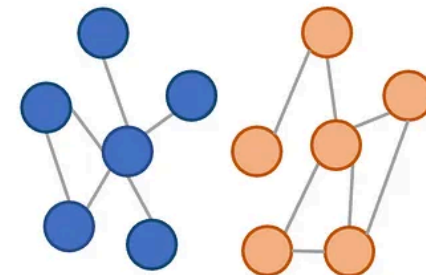
Node Classification



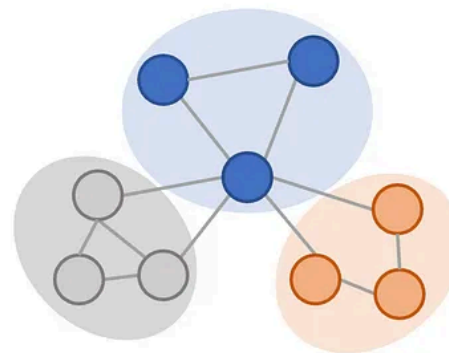
Link Prediction



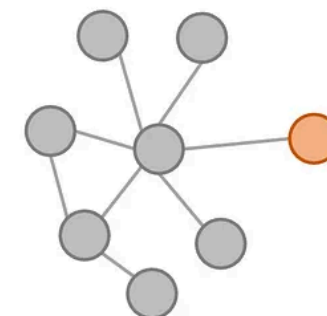
Graph Classification



Community Detection



Anomaly Detection



Metodi Tradizionali (Shallow)

Nell'ambito dello **"Statistical Relational Learning"** diversi approcci tradizionali sono stati sviluppati:

- **Logic Programming**
- **Inductive Logic Programming** (ILP)
- **Rule Mining**: AMIE
- **Graphical Models** Conditional Random Fields, Modelli Relazionali Probabilistico, Reti di Markov, Reti di dipendenza, etc etc

Limitazioni

Scalabilità, debole potere di modellazione, approcci non differenziabili

Metodi Tradizionali (Deep)

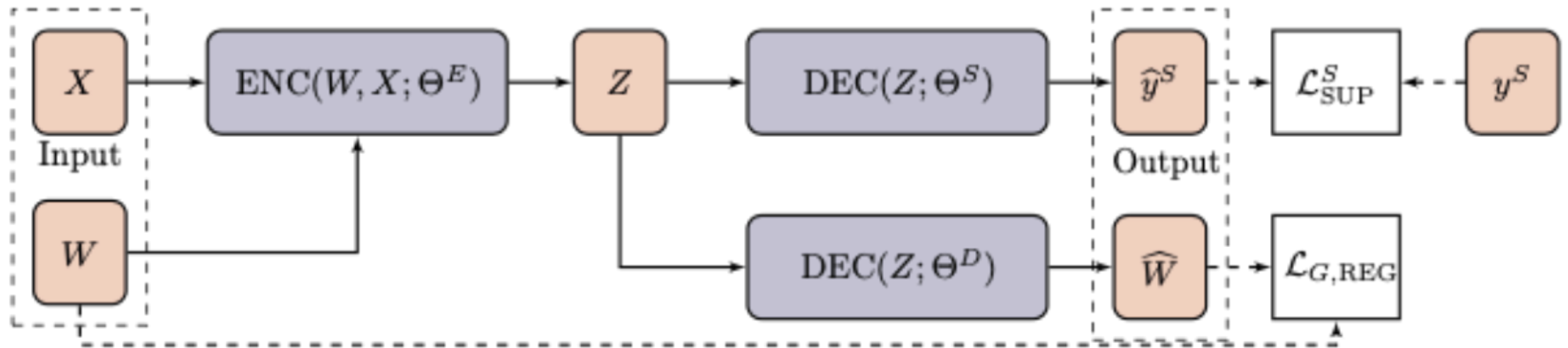
- Le **CNN** lavorano naturalmente su "griglie", quali immagini
- Le **RNN** lavorano su sequenze, quali testo

Abbiamo la necessità di gestire la **complessità** dei grafi:

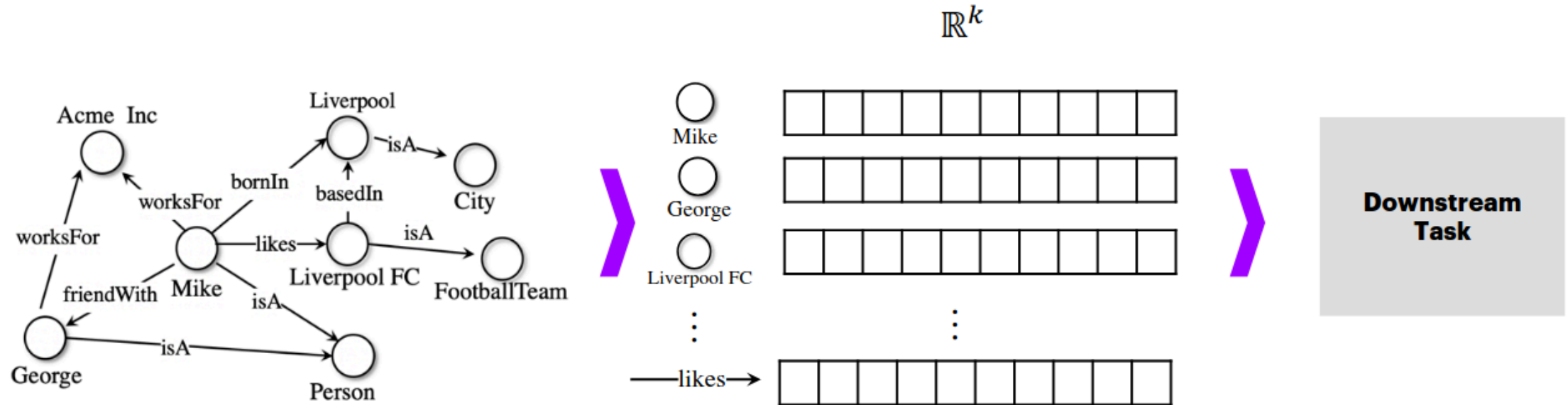
- Assenza di **spatial locality**
- Assenza di un **ordinamento dei nodi** fisso
- Naturalmente **multimodali**, concetti, testi, numeri, date, immagini

Apprendimento su Grafi come Encoder - Decoder

- Nei modelli che vedremo, il decoder prenderà la forma di una **lookup table**



Apprendere le Rappresentazioni di Nodi e Relazioni



Metodologie basate sull'apprendimento automatico di **rappresentazioni vettoriali** di nodi e relazioni. Proiettiamo la conoscenza **simbolica** su uno spazio vettoriale continuo.

Da nodi e relazioni...

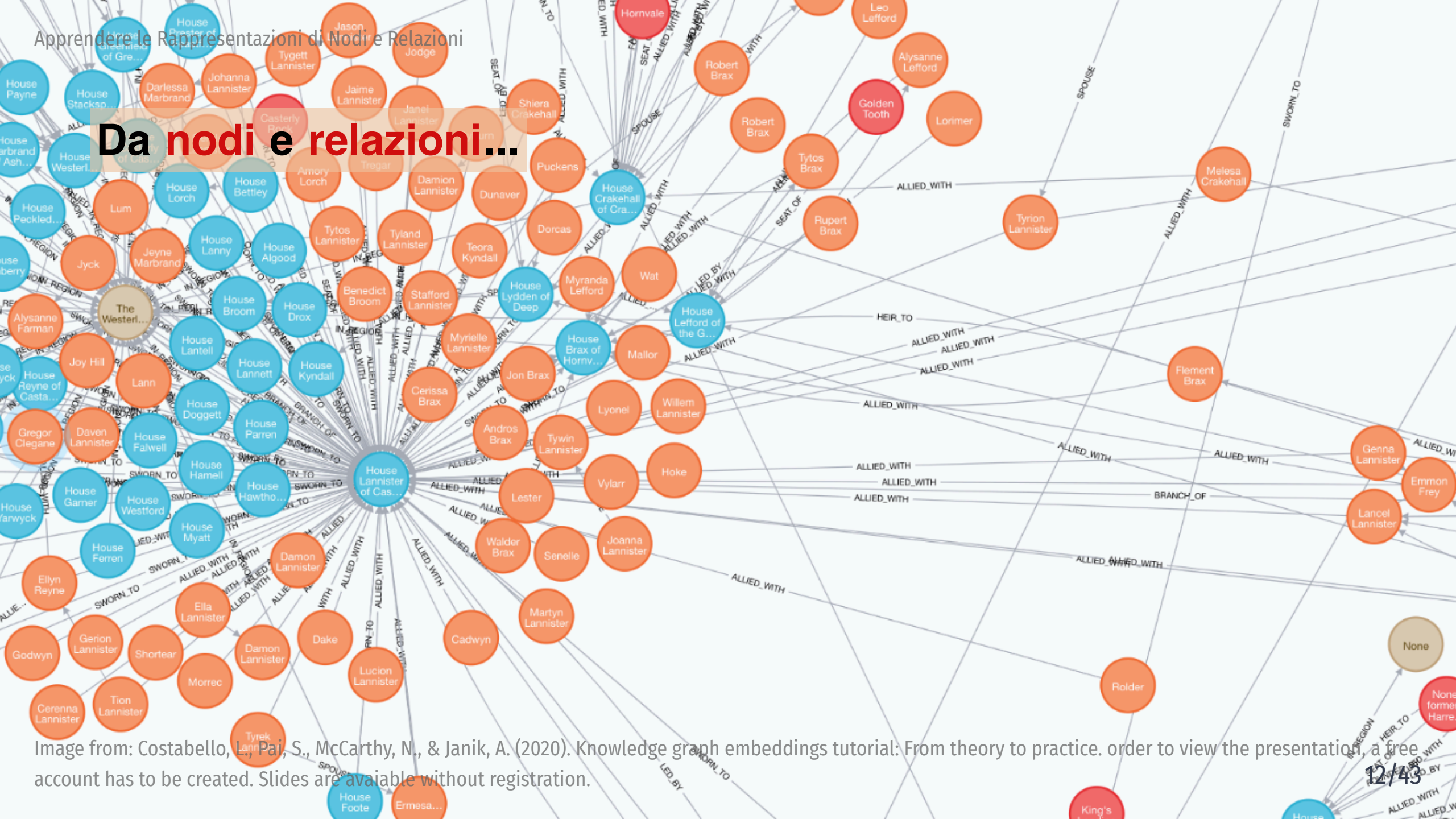
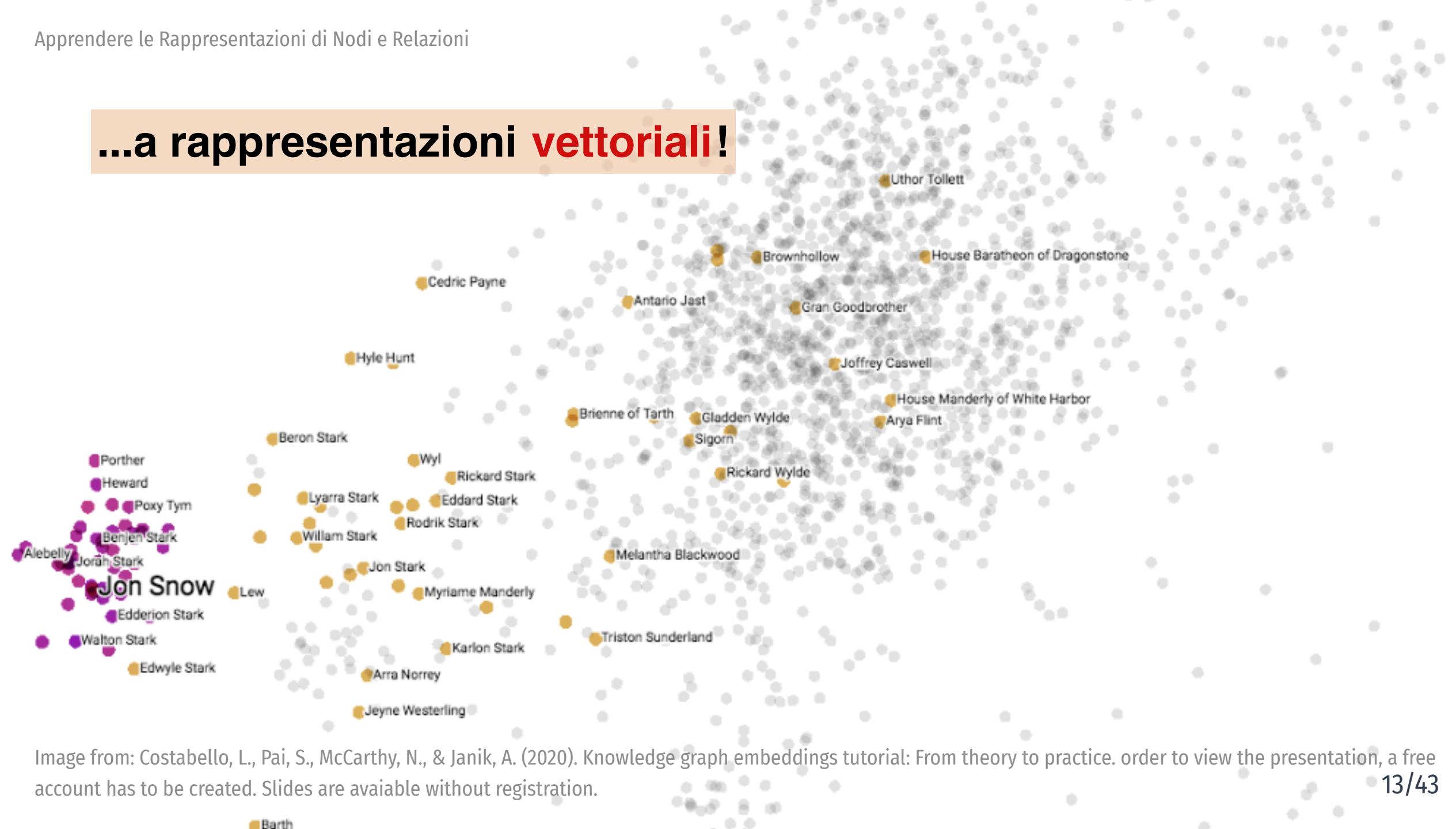


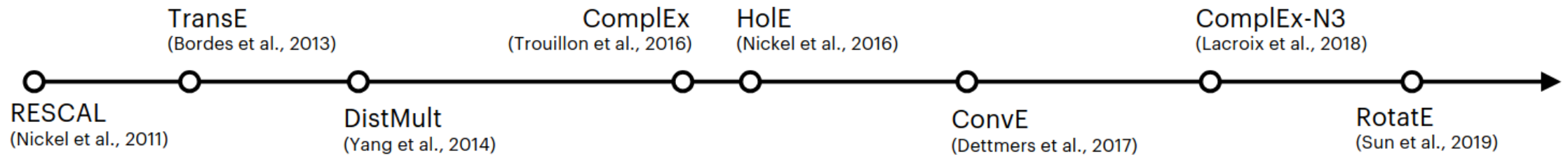
Image from: Costabello, L., Pai, S., McCarthy, N., & Janik, A. (2020). Knowledge graph embeddings tutorial: From theory to practice. order to view the presentation, a free account has to be created. Slides are available without registration.

...a rappresentazioni **vettoriali**!



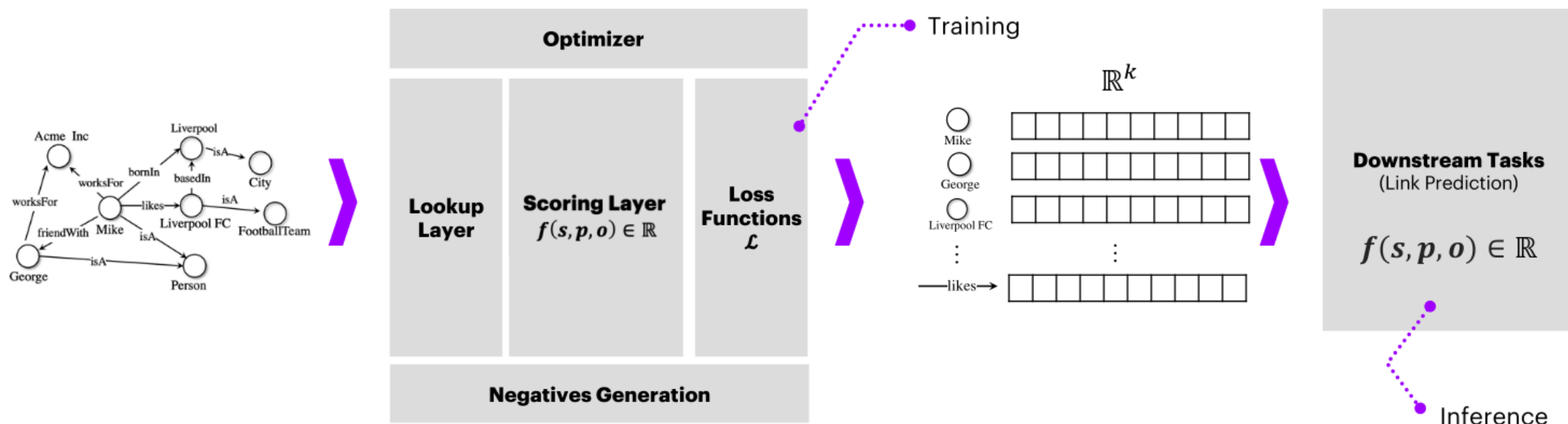
Modelli in Letteratura

(Some) KGE models in recent published literature:



In letteratura troviamo varie tipologie di modelli, che si possono categorizzare sulla base della tipologia di apprendimento, **translational**, **factorization**, **neural**, etc etc

Anatomia di un modello di KGE



- Funzione di **score** delle triple $f(t)$
- Funzione di **loss** \mathcal{L}
- Strategia di generazione dei **negativi**

Funzione di **score** delle triple

La funzione f assegna uno score ad una tripla $\langle s, p, o \rangle$. La strategia di scoring varia in base alla metodologia scelta, ad esempio:

Traslation Based

TransE

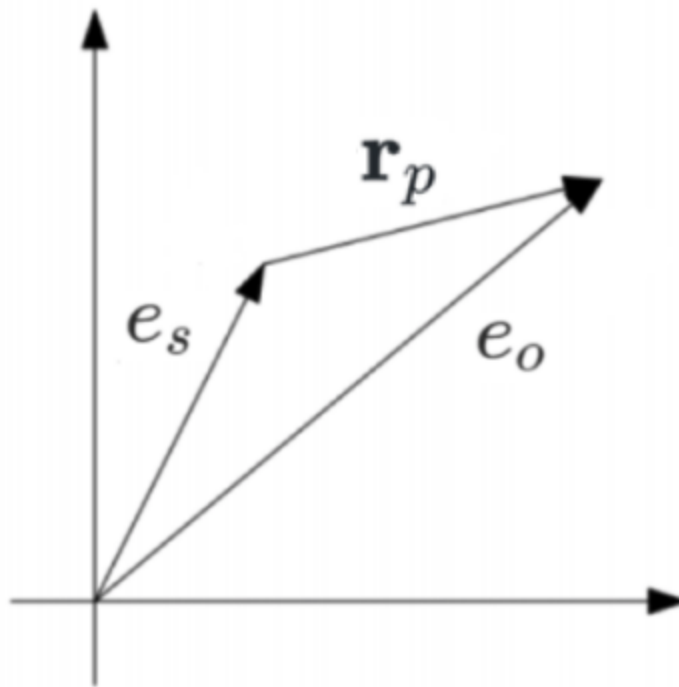
$$f_{TransE}(\langle s, p, o \rangle) = -\|(e_s - r_p) - e_o\|$$

RotatE

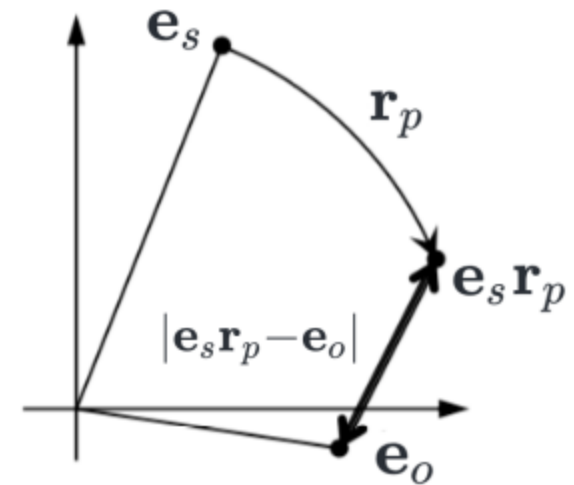
$$f_{RotatE}(\langle s, p, o \rangle) = -\|e_s \circ r_p - e_o\|_n$$

Traslation - Esempio grafico

TransE



RotatE



Factorization Based

RESCAL

$$f_{RESCAL}(\langle s, p, o \rangle) = e_s^\top W_r e_o$$

Convolution Based

ConvE

$$f_{ConvE}(\langle s, p, o \rangle) = f(\text{vec}(f([\overline{e_s}; \overline{r_p}] * \omega)) \mathbf{W}) e_o$$

con f funzione non lineare, $*$ operatore di convoluzione e \overline{x} un reshape 2D del vettore.

Complessità e Proprietà dei KGE

Modelli si differenziano nel tipo di **spazio** utilizzato (vedremo più avanti), funzione di scoring, complessita nello spazio e nel tempo e tipo di **proprietà che possono modellare**

Method	Publisher	Math Space	Time Complexity	Space Complexity
SE [15]	AAAI 2011	Euclidean	$\mathcal{O}(d^2)$	$\mathcal{O}(N_e d + N_r d^2)$
RESCAL [130]	ICML 2011	Euclidean	$\mathcal{O}(d^2)$	$\mathcal{O}(N_e d + N_r d^2)$
TransE [14]	NeurIPS 2013	Euclidean	$\mathcal{O}(d)$	$\mathcal{O}(N_e d + N_r d)$
TransH [190]	AAAI 2014	Euclidean	$\mathcal{O}(d)$	$\mathcal{O}(N_e d + N_r d)$
TransR [109]	AAAI 2015	Euclidean	$\mathcal{O}(dk)$	$\mathcal{O}(N_e d + N_r dk)$
TransD [75]	ACL-IJCNLP 2015	Euclidean	$\mathcal{O}(\max(d, k))$	$\mathcal{O}(N_e d + N_r k)$
DistMult [209]	ICLR 2015	Euclidean	$\mathcal{O}(d)$	$\mathcal{O}(N_e d + N_r d)$
ComplEx [168]	ICML 2016	Euclidean	$\mathcal{O}(d)$	$\mathcal{O}(N_e d + N_r d)$
TranSparse [76]	AAAI 2016	Euclidean	$\mathcal{O}(dk)$	$\mathcal{O}(N_e d + (1 - \theta) N_r dk)$
HolE [129]	AAAI 2016	Euclidean	$\mathcal{O}(d \log d)$	$\mathcal{O}(N_e d + N_r d)$
ANALOGY [110]	ICML 2017	Euclidean	$\mathcal{O}(d)$	$\mathcal{O}(N_e d + N_r d)$
ConvE [33]	AAAI 2018	Euclidean	$\mathcal{O}(d)$	$\mathcal{O}(N_e d + N_r d)$
RotE [24]	ACL 2020	Euclidean	$\mathcal{O}(d)$	$\mathcal{O}(N_e d + 2 N_r d)$
Rot2L [179]	ACL Findings 2021	Euclidean	$\mathcal{O}(d)$	$\mathcal{O}(N_e d + 2(N_r + 5)d)$
ItôE _ℝ [124]	ACL Findings 2023	Euclidean	$\mathcal{O}(d)$	$\mathcal{O}(N_e d + N_r k)$
RotH [24]	ACL 2020	Hyperbolic	$\mathcal{O}(d)$	$\mathcal{O}(N_e d + 3(N_r + 1)d)$
ItôE _ℙ [124]	ACL Findings 2023	Hyperbolic	$\mathcal{O}(d)$	$\mathcal{O}(N_e d + N_r k)$
ManifoldE _ℳ [198]	IJCAI 2016	Spherical	$\mathcal{O}(d)$	$\mathcal{O}(N_e d + N_r d)$
ItôE _ℳ [124]	ACL Findings 2023	Spherical	$\mathcal{O}(d)$	$\mathcal{O}(N_e d + N_r k)$
UltraE [204]	KDD 2022	Ultrahyperbolic	$\mathcal{O}(d)$	$\mathcal{O}(N_e d + N_r d)$

Table from: Cao, J., Fang, J., Meng, Z., & Liang, S. (2024). Knowledge graph embedding: A survey from the perspective of representation spaces. ACM Computing Surveys, 56(6), 1-42.

Complessità e Proprietà dei KGE

- Simmetria <Alice, marriedTo, Bob>
- Asimmetria <Alice, childOf, Jack>
- Inversione <Alice,childOf,Jack> , <Jack,fatherOf, Alice>
- Composizione <Alice, childOf, Jack> , <Jack, siblingOf, Mary> , <Alice, nieceOf, Mary>

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

Funzione di **loss** \mathcal{L}

Varie funzioni di loss provenienti dagli ambiti del machine learning e information retrieval (**basate su ranking**)

Pairwise Margin-Based Hinge Loss

$$\mathcal{L}(\Theta) = \sum_{t^+ \in \mathcal{G}} \sum_{t^- \in \mathcal{C}} \max(0, [\gamma + f(t^-; \Theta) - f(t^+; \Theta)])$$

Negative Log-Likelihood / Cross Entropy

$$\mathcal{L}(\Theta) = \sum_{t \in \mathcal{G} \cup \mathcal{C}} \log(1 + \exp(-y f(t; \Theta)))$$

$y \in \{-1, +1\}$ label della tripla t

Ulteriori funzioni di **loss**

Binary Cross-Entropy

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

Self-Adversarial

$$\mathcal{L}(\Theta) = -\log \sigma(\gamma + f(t^+; \Theta)) - \sum_{t \in \mathcal{G}} p(t^-; \Theta) \log \sigma(-f(t^-; \Theta) - \gamma)$$

$p(t^-; \Theta)$ peso del negativo t

Strategia di generazione dei negativi

Prende anche il nome di **negative sampling**.

Come possiamo generare negativi (fatti falsi)?

Local Closed World Assumption: Un KG è solo localmente completo

Andiamo a **corrompere** le triple creando dei negativi sintetici:

$$\mathcal{C} = \{ \langle \hat{s}, p, o \rangle \mid \hat{s} \in \mathcal{E} \} \cup \{ \langle s, p, \hat{o} \rangle \mid \hat{o} \in \mathcal{E} \}$$

Esempio di negative sampling

$$\mathcal{E} = \{\text{Mike}, \text{Liverpool}, \text{AcmeInc}, \text{George}, \text{LiverpoolFC}\}$$

$$\mathcal{R} = \{\text{bornIn}, \text{friendWith}\}$$

$$t \in \mathcal{G} = \langle \text{Mike} \text{ bornIn } \text{Liverpool} \rangle$$

$$\mathcal{C}_t = \left\{ \begin{array}{l} \langle \text{Mike} \text{ bornIn } \text{AcmeInc} \rangle, \\ \langle \text{Mike} \text{ bornIn } \text{LiverpoolFC} \rangle, \\ \langle \text{George} \text{ bornIn } \text{Liverpool} \rangle, \\ \langle \text{AcmeInc} \text{ bornIn } \text{Liverpool} \rangle \end{array} \right\}$$

Altri dettagli supplementari

- **Ottimizzazione**: Qualsiasi algoritmo che avete già studiato nel corso, SGD, ADAM, etc
- **Inizializzazione**: Il processo deve partire da embedding iniziali, si possono utilizzare funzioni quali la distribuzione **unioforme**, **normale** o **xavier**
- **Regolarizzazione**: L1,L2, Dropout, Vincoli di **Norma**

Valutazione e Metriche

Nella link prediction impariamo uno score proporzionale alla probabilità che la tripla sia vera, in altre parole stiamo imparando un task di **ranking** delle triple.

Possiamo prendere "in prestito" le metriche di valutazione dell'**information retrieval**. Non abbiamo una bisogno di una **ground truth** nella valutazione del test set. Rispondiamo alla domanda:

"Quanto in alto vengono **classificate** le triple positive rispetto alle triple negative generate artificialmente?"

! In **media**, che **rank** viene assegnato alla tripla vera rispetto ai negativi sintetici?

Mean Rank (MR)

$$MR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} rank(\langle s, p, o \rangle_i)$$

Mean Reciprocal Rank (MRR)

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank(\langle s, p, o \rangle_i)}$$

! ? Quante volte la tripla vera è stata posizionata almeno nelle **prime N posizioni**?

Hits@N

$$Hits@N = \sum_{i=1}^{|Q|} 1(rank(\langle s, p, o \rangle_i) \leq N)$$

Esempio

s	p	o	score	rank
Mike	bornIn	Leeds	0.78	1
Mike	bornIn	Italy	0.75	2
Mike	bornIn	Germany	0.69	3
George	bornIn	Italy	0.45	4
Mike	bornIn	George	0.23	5

s	p	o	score	rank
Mike	friendWith	George	0.90	1
Mike	friendWith	Jim	0.34	2
Acme	friendWith	George	0.29	3
Mike	friendWith	Italy	0.20	4
France	friendWith	George	0.15	5

$MR = 1.5$ $MRR = .75$ $Hits@1 = .5$ $Hits@3 = 1.0$

Argomenti **Avanzati**:

1. **Calibrazione**
2. KGE & Ragionamento **Neuro-Simbolico**
3. **Spazi** di Embedding

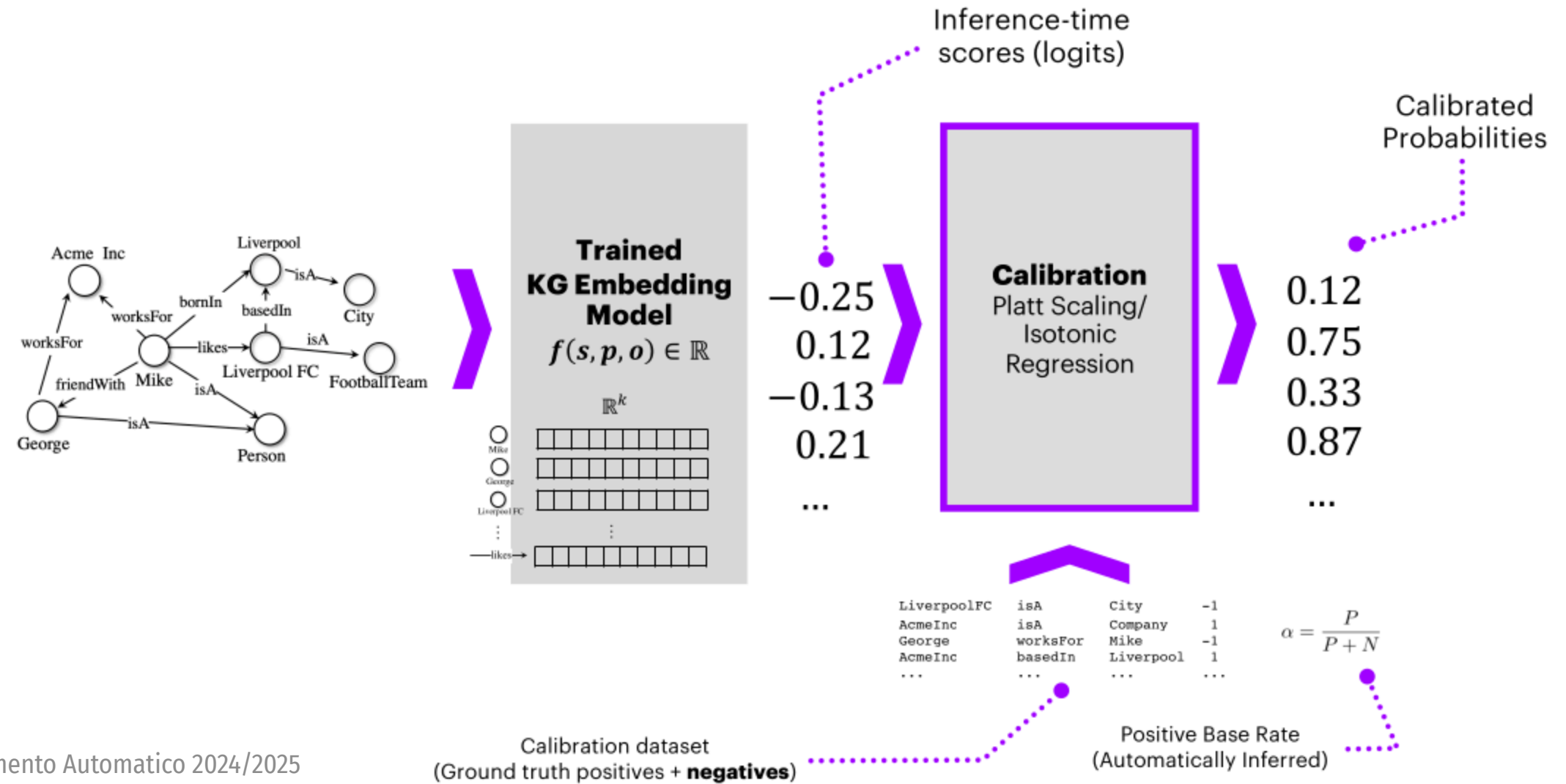
Calibrazione degli score

Gli score generati dai modelli KGE sono **non calibrati** (gli score non corrispondono a probabilità o sono affidabili di correttezza).

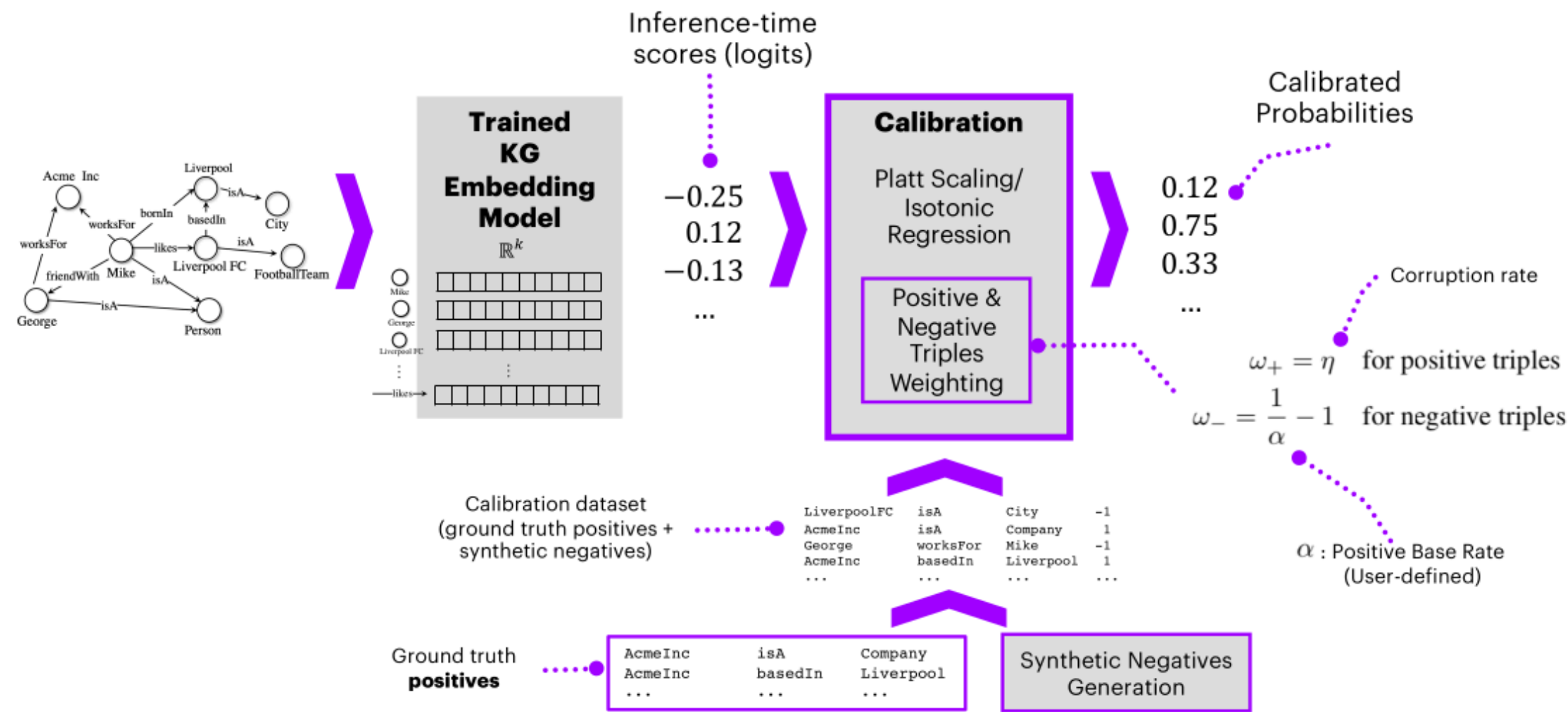
La calibrazione è un processo di **post-processing** che trasforma gli score in effettive **probabilità**! Ci troviamo davanti a due scenari:

- **CWA**: Abbiamo a disposizione dei negativi esplicitamente taggati
- **OWA**: Non abbiamo dei negativi, dobbiamo pesare le triples

Calibrazione con Negativi Ground Truth CWA

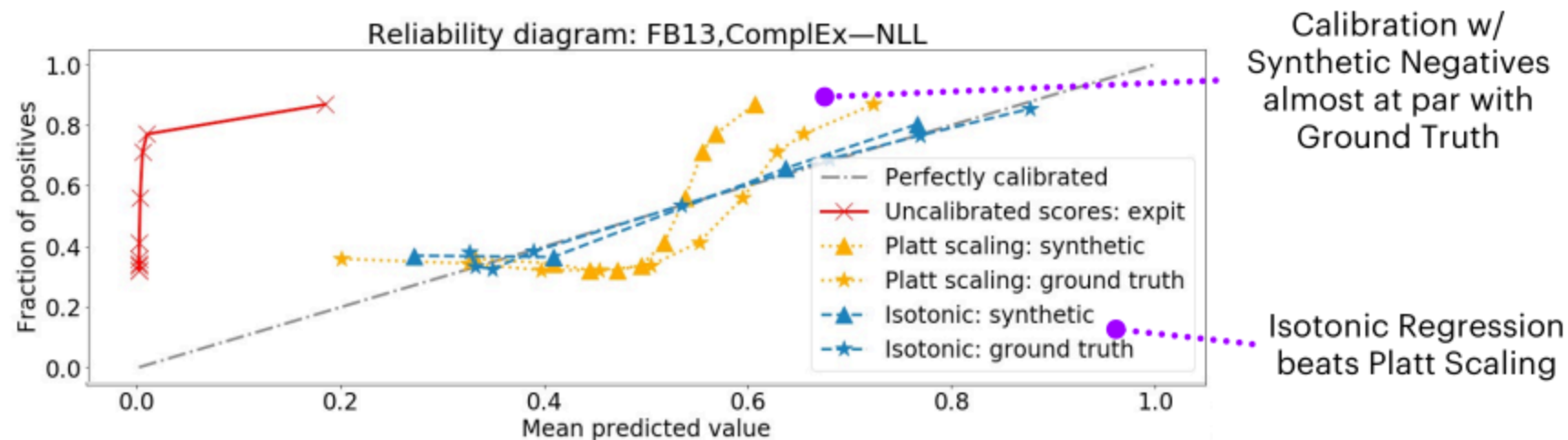


Calibrazione Senza Negativi Ground Truth OWA



Calibrazione Efficacia

Modelli calibrati producono risultati migliori rispetto a modelli non calibrati, producono inoltre predizioni più **affidabili** e **interpretabili**



KGE & Ragionamento **Neuro-Simbolico**

Iniettare conoscenza di background nei modelli tramite "**Soft Constraints**". Avendo a disposizione una ontologia, possiamo iniettare **regole** nella loss function:

$$\begin{aligned} \text{hasNeighbor} &\equiv \text{hasNeighbor}^- & \text{isMarriedTo} &\equiv \text{isMarriedTo}^- \\ \text{playsFor} &\equiv \text{isAffiliatedTo}^- & \text{isConnectedTo} &\equiv \text{isConnectedTo}^- \end{aligned}$$

Prendiamo in esempio regole di **equivalenza**

Loss Regularizzata:

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

Regularizzazione:

$$\mathcal{R}_{\mathcal{S}}(\Theta) = \sum_{p \equiv q \in \mathcal{A}_1} D[r_p || r_q] + \sum_{p \equiv q^- \in \mathcal{A}_2} D[r_p || \Phi(r_q)]$$

- $D[x || y] = \|x - y\|_2^2$ Misura divergenza
- Φ Trasformazione dipendente dal modello
- \mathcal{A}_1 Insieme assiomi **equivalenza**
- \mathcal{A}_2 Insieme assiomi **inverse**

KGE & Ragionamento **Neuro-Simbolico**

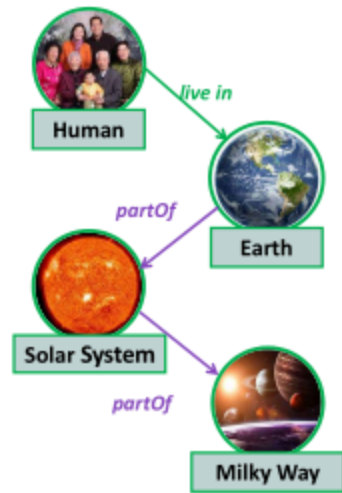
Andiamo quindi ad unire **rule-based models** e **KGE**, sfruttiamo sia la potenza di generalizzazione e scalabilità dei KGE, con l'interpretabilità dei modelli basati su regole.

Ulteriore metodologie sono i **Neural Theorem Provers** (NTP) che implementano il ragionamento in architetture completamente **differenziabili**

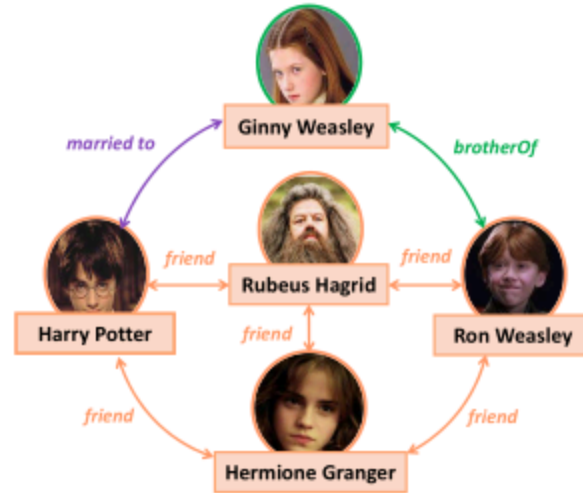
Spazi e Metodi KGE

Spazi più complessi possono essere utilizzati per:

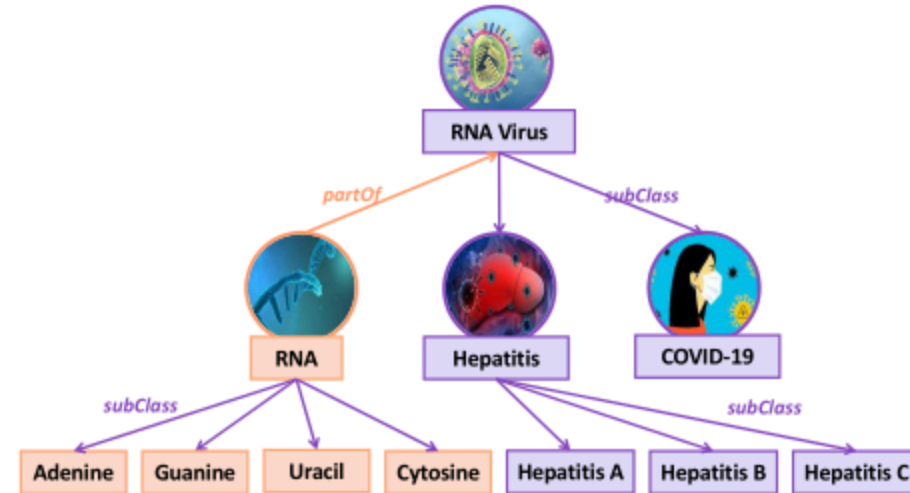
- Preservare **pattern relazionali** tra le entità
- Preservare **pattern strutturali** (gerarchie)
- Modellare **proprietà** desiderate, sfruttando le proprietà matematiche degli spazi



(a) Chain structure.



(b) Ring structure.



(c) Hierarchy structure.

Spazi e Metodi KGE: Strutture Algebriche

- **Spazi Vettoriali**: Sfruttano somma tra vettori e moltiplicazione per scalare per proiettare entità e relazioni nello stesso spazio. Preserva le interazioni relazionali.
- **Gruppi**: Permettono di modellare proprietà come inversione e composizione
- **Anelli**: Utili nella rappresentazione con task orientatin-related, grazie alla sua superfice non orientata

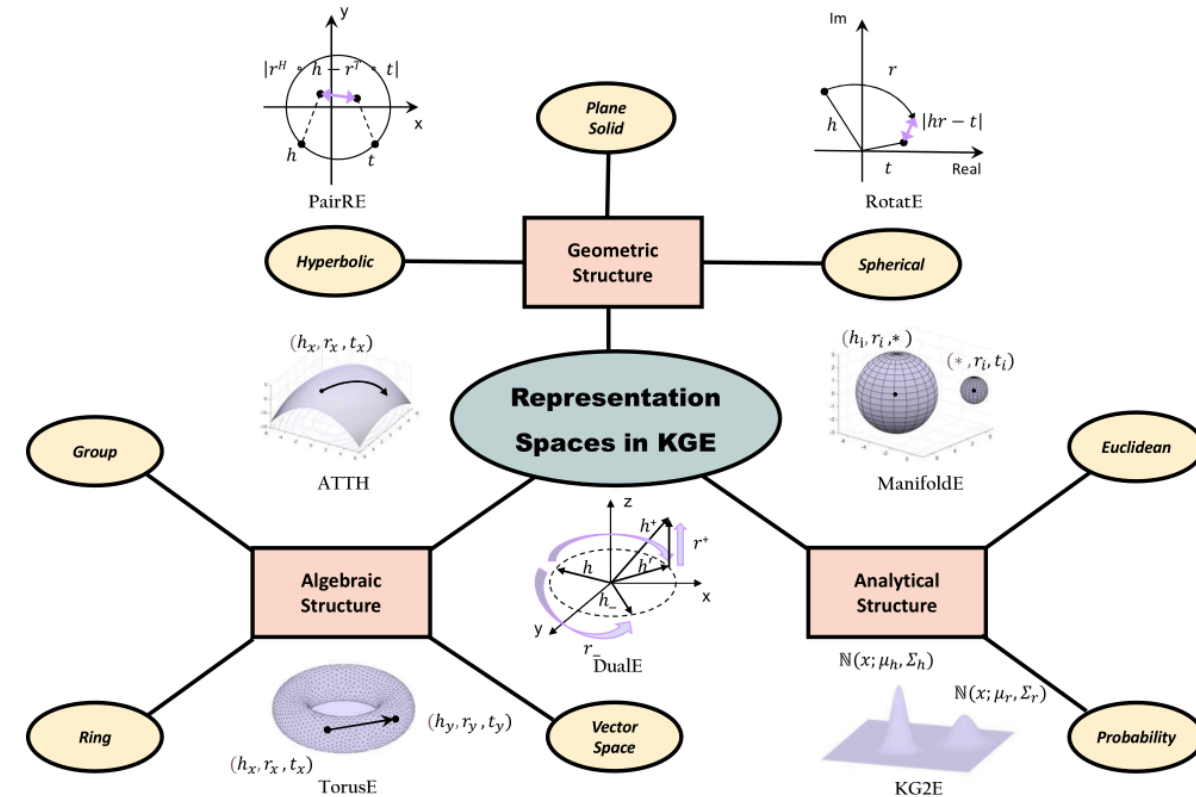


Table from: Cao, J., Fang, J., Meng, Z., & Liang, S. (2024). Knowledge graph embedding: A survey from the perspective of representation spaces. ACM Computing Surveys, 56(6), 1-42.

Spazi e Metodi KGE: Strutture Geometriche

- **Euclidea**: Efficace rappresentazione di pattern relazionali
- **Iperbolica**: Spazi utili nel modellare strutture gerarchiche.
- **Sferica**: Eccellenti nel modellare strutture ad anello grazie alla loro natura circolare.

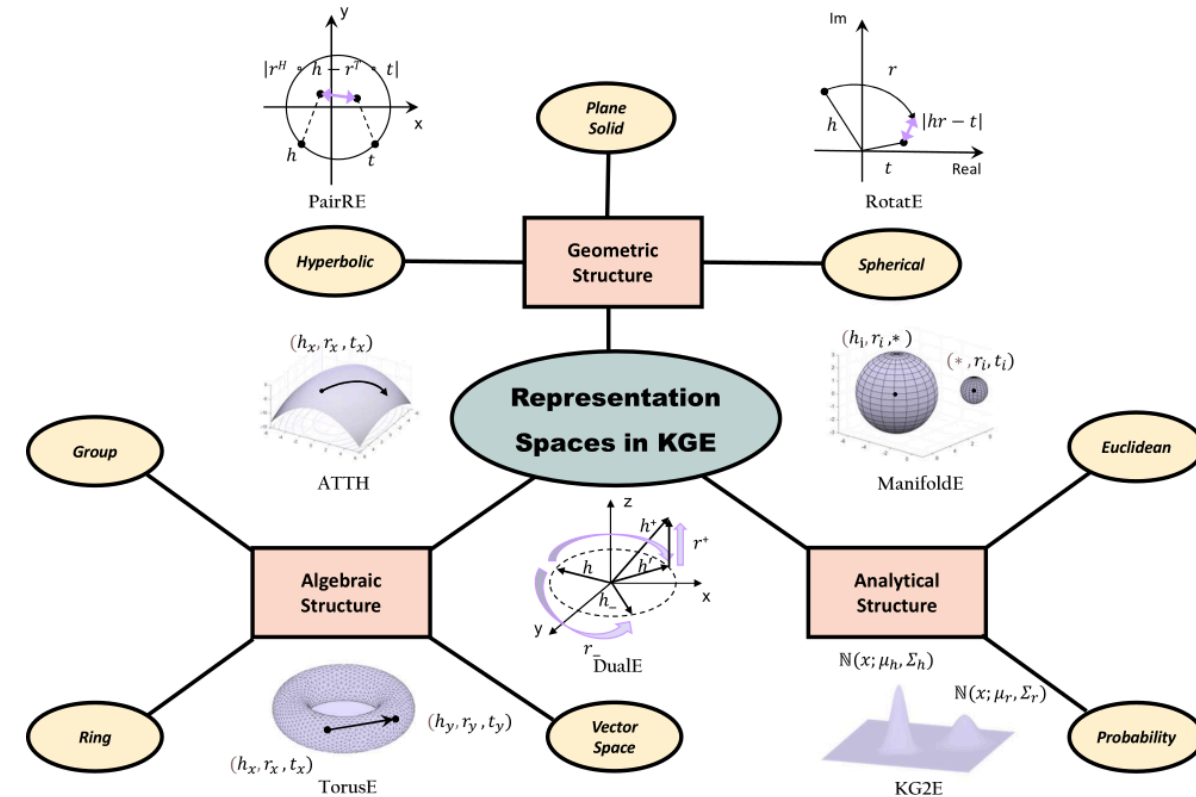


Table from: Cao, J., Fang, J., Meng, Z., & Liang, S. (2024). Knowledge graph embedding: A survey from the perspective of representation spaces. ACM Computing Surveys, 56(6), 1-42.

Spazi e Metodi KGE: Strutture Analitiche

- **Probabilità**: Capaci di modellare conoscenza incerta
- **Euclideo**: Facilitano l'acquisizione di rappresentazioni dinamiche e continue

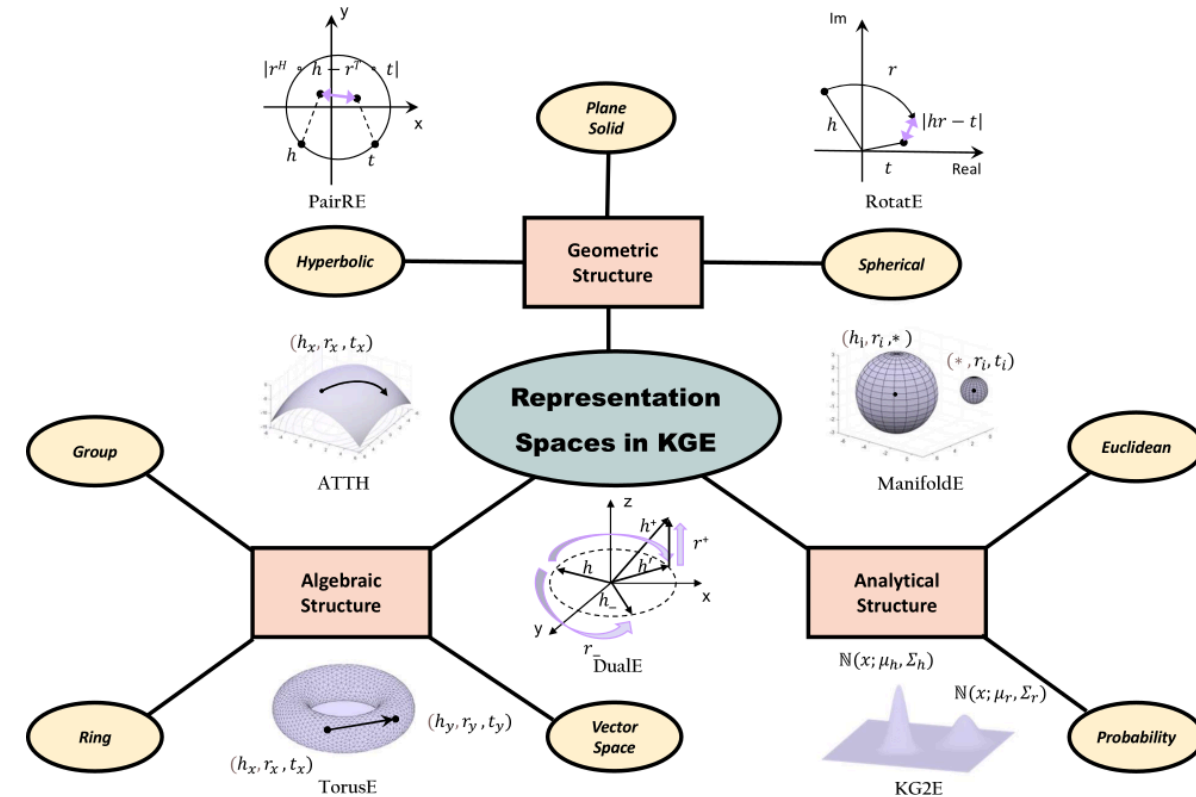


Table from: Cao, J., Fang, J., Meng, Z., & Liang, S. (2024). Knowledge graph embedding: A survey from the perspective of representation spaces. ACM Computing Surveys, 56(6), 1-42.



Grazie per l'attenzione

Ivan Diliso, Ph.D Student, ARA

Riferimenti

Il materiale è un riadattamento delle slide presentate durante il corso "Semantic Technologies and Knowledge Graphs" del Corso di Laurea Magistrale in Artificial Intelligence. Trovate ulteriori dettagli in:

- **KGE e**: Costabello, L., Pai, S., McCarthy, N., & Janik, A. (2020). Knowledge graph embeddings tutorial: From theory to practice. order to view the presentation.
- **Spazi oltre l'euclideo**: Cao, J., Fang, J., Meng, Z., & Liang, S. (2024). Knowledge graph embedding: A survey from the perspective of representation spaces. ACM Computing Surveys, 56(6), 1-42