Machine Learning and Knowledge Graphs

Pasquale Minervini
University College London
@pminervini

Outline

- **Knowledge Graphs**
  - What are they?
  - Where are they?
  - Where do they come from?

Outline

- **Knowledge Graphs**
  - What are they?
  - Where are they?
  - Where do they come from?
- **Statistical Relational Learning in Knowledge Graphs**
  - Explainable Models (Observable FMs)
  - Black-Box Models (Latent FMs)
  - Towards Combining the Two Worlds

Outline

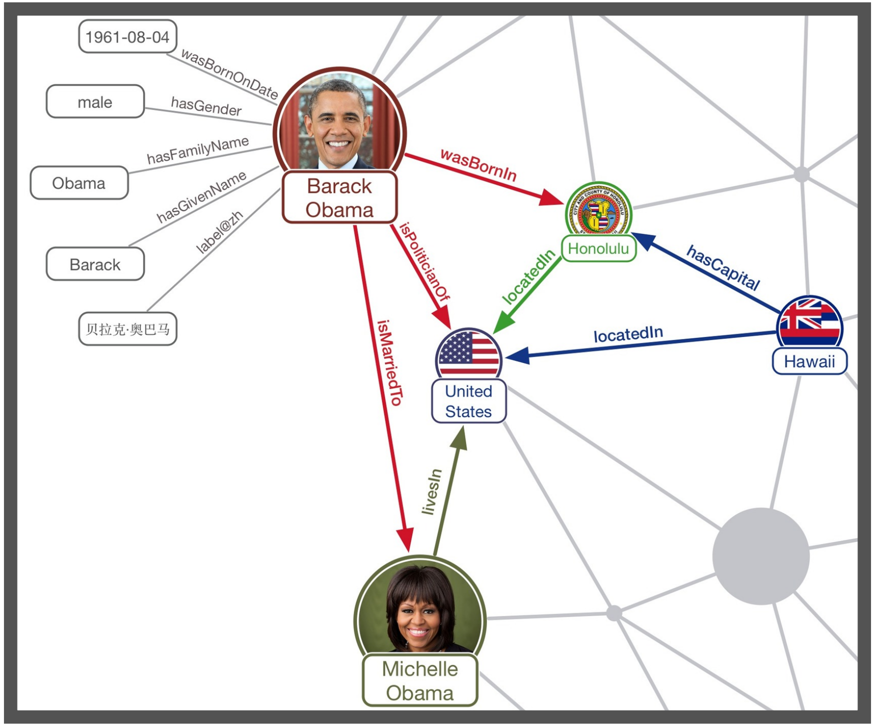
- **Knowledge Graphs**
  - What are they?
  - Where are they?
  - Where do they come from?
- **Statistical Relational Learning in Knowledge Graphs**
  - Explainable Models (Observable FMs)
  - Black-Box Models (Latent FMs)
  - Towards Combining the Two Worlds
- **Differentiable Reasoning**

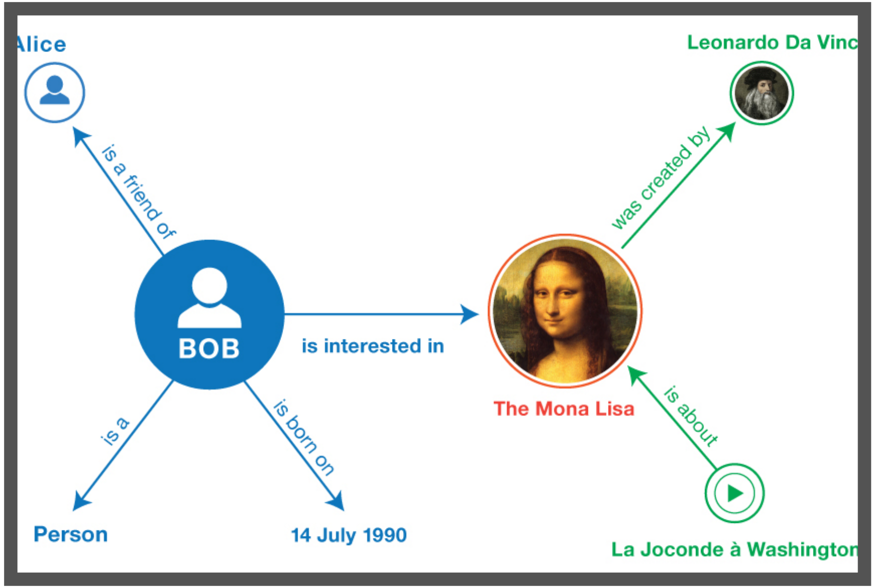
Knowledge Graphs

Knowledge Graphs are *graph-structured Knowledge Bases*, where knowledge is encoded by *relationships between entities.*

Knowledge Graphs

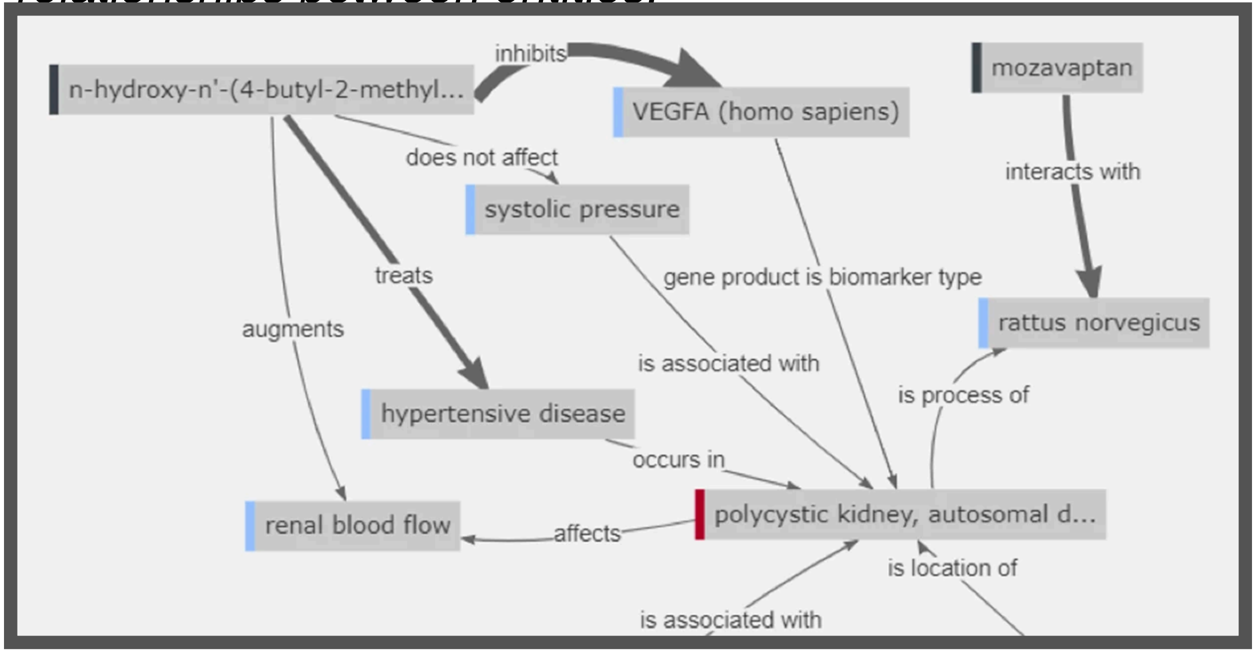
Knowledge Graphs are *graph-structured Knowledge Bases*, where knowledge is encoded by *relationships between entities.*





Knowledge Graphs

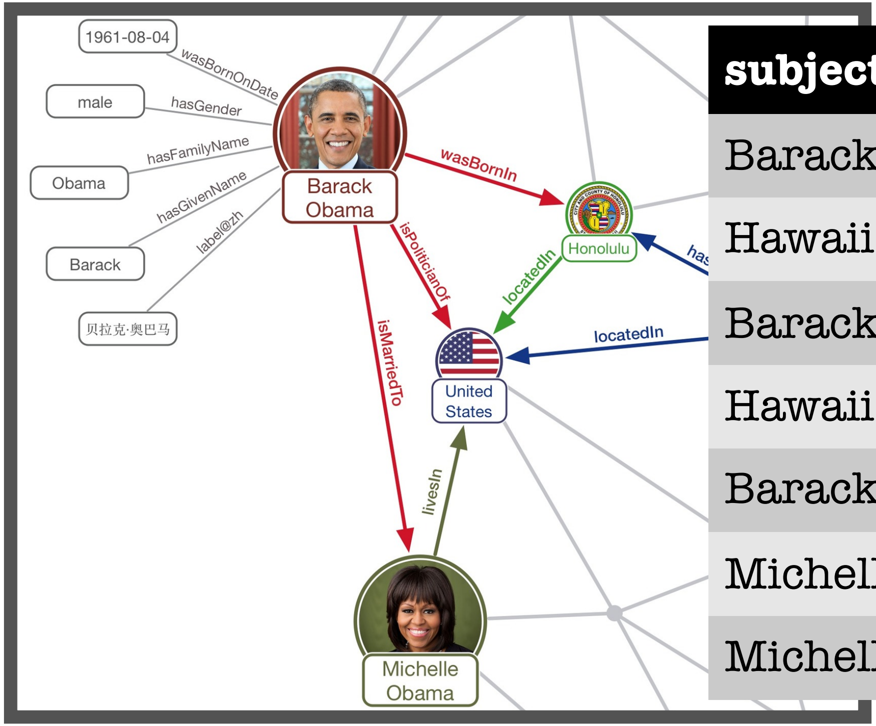
Knowledge Graphs are *graph-structured Knowledge Bases*, where knowledge is encoded by *relationships between entities.*



Drug Prioritization using the semantic properties of a Knowledge Graph, Nature 2019

Knowledge Graphs

Knowledge Graphs are *graph-structured Knowledge Bases*, where knowledge is encoded by *relationships between entities.*



subject	predicate	object
Barack Obama	was born in	Honolulu
Hawaii	has capital	Honolulu
Barack Obama	is politician of	United States
Hawaii	is located in	United States
Barack Obama	is married to	Michelle Obama
Michelle Obama	is a	Lawyer
Michelle Obama	lives in	United States

Industry-Scale Knowledge Graphs

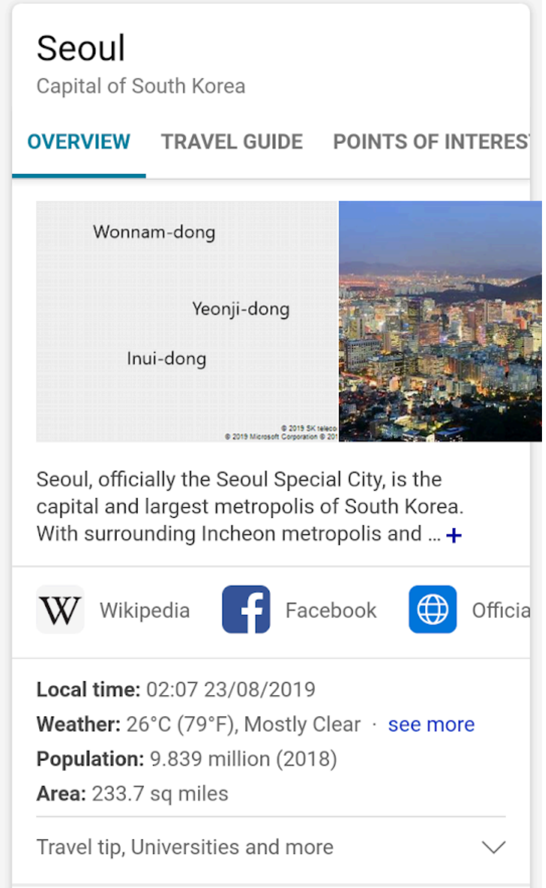
In many enterprises, Knowledge Graphs are **critical** — they provide structured data and factual knowledge that drives many products, making them more “intelligent”.

Industry-Scale Knowledge Graphs in Microsoft

In *Microsoft* there are several major graph systems used by products:

- *Bing Knowledge Graph* — contains information about the world and powers question answering services on Bing.
- *Academic Graph* — collection of entities such as people, publications, felds of study, conferences, etc. and helps users discovering relevant research works.
- *LinkedIn Graph* — contains entities such as people, jobs, skills, companies, etc. and it is used to find economy-level insights for countries and regions.

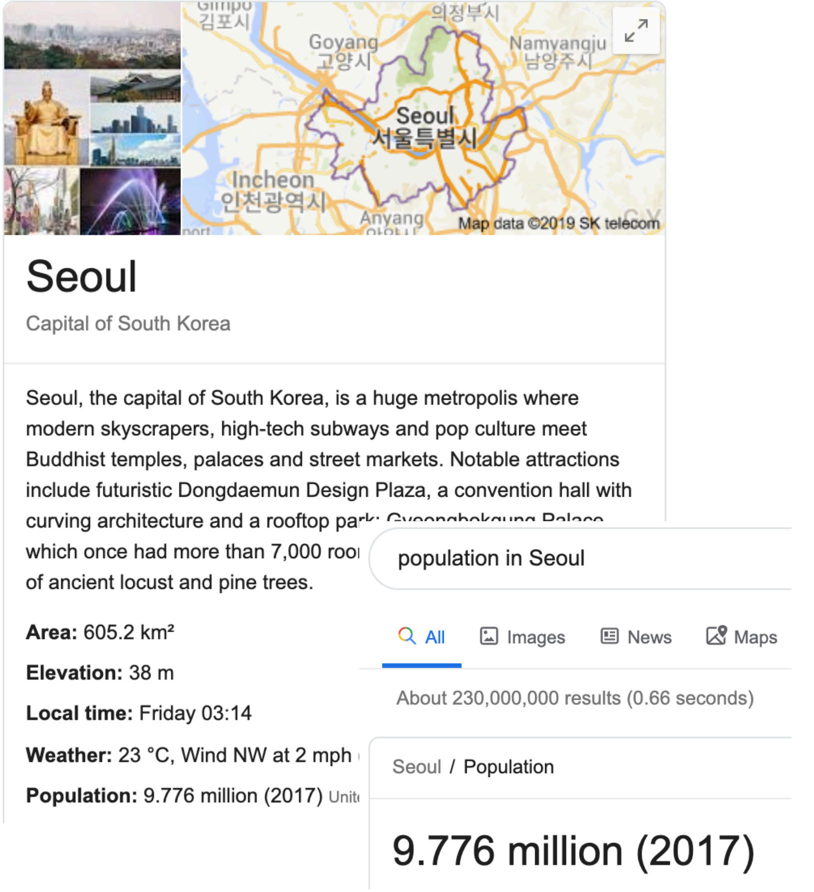
~2 Billion primary entities, ~55 Billion Facts



Industry-Scale Knowledge Graphs in Google

The *Google Knowledge Graph* contains more than 70 billion assertions describing a billion entities and covers a variety of subject matter — "things not strings".

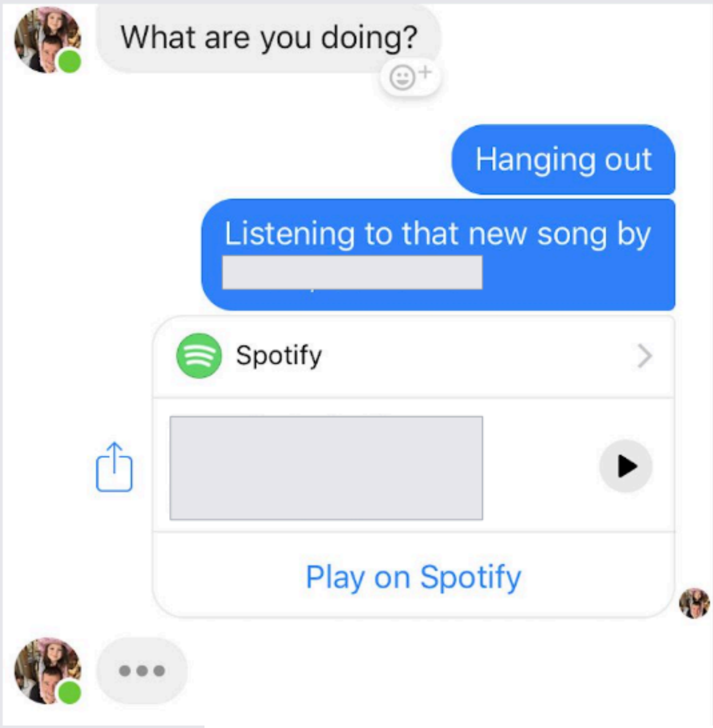
Used for answering factoid queries about entities served from the Knowledge Graph.



1 Billion entities, ~70 Billion assertions

Industry-Scale Knowledge Graphs in Facebook

World's largest social graph — *Facebook's Knowledge Graph* focuses on socially relevant entities, such as celebrities, places, movies, and music. Used to *recommend smart replies*, *entity detection*, and *easy sharing*.

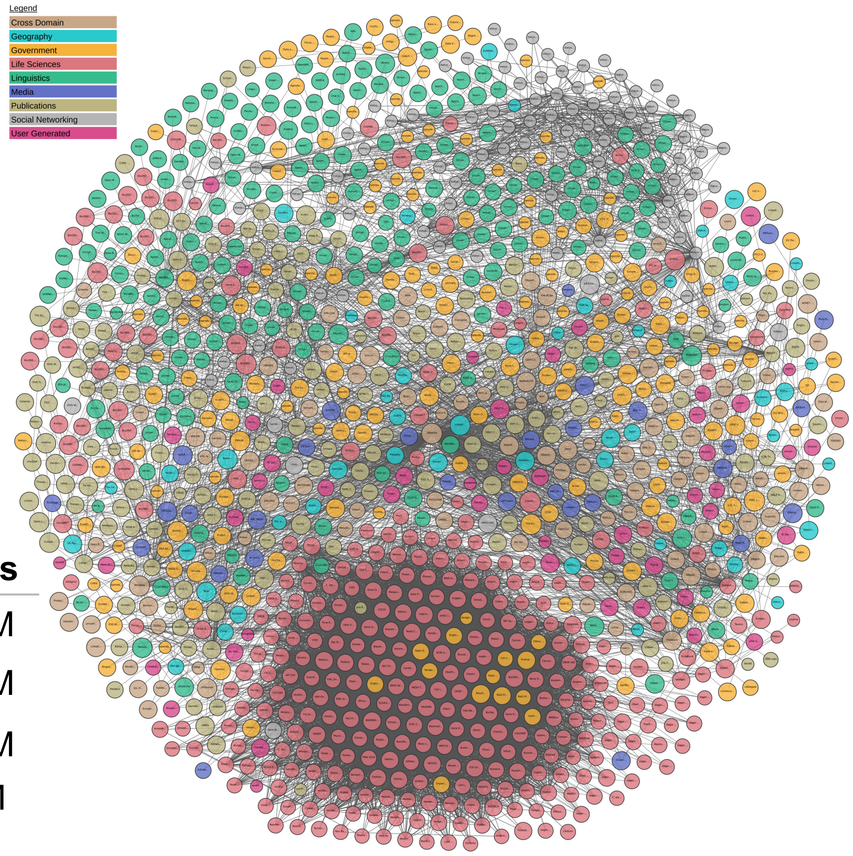


~50 mllion primary entities, ~500 million assertions

The Linked Open Data Cloud

Linked Open Data cloud - over 1200 interlinked KGs encoding more than 200M facts about more than 50M entities.

Spans a variety of domains, such as Geography, Government, Life Sciences, Linguistics, Media, Publications, and Cross-domain



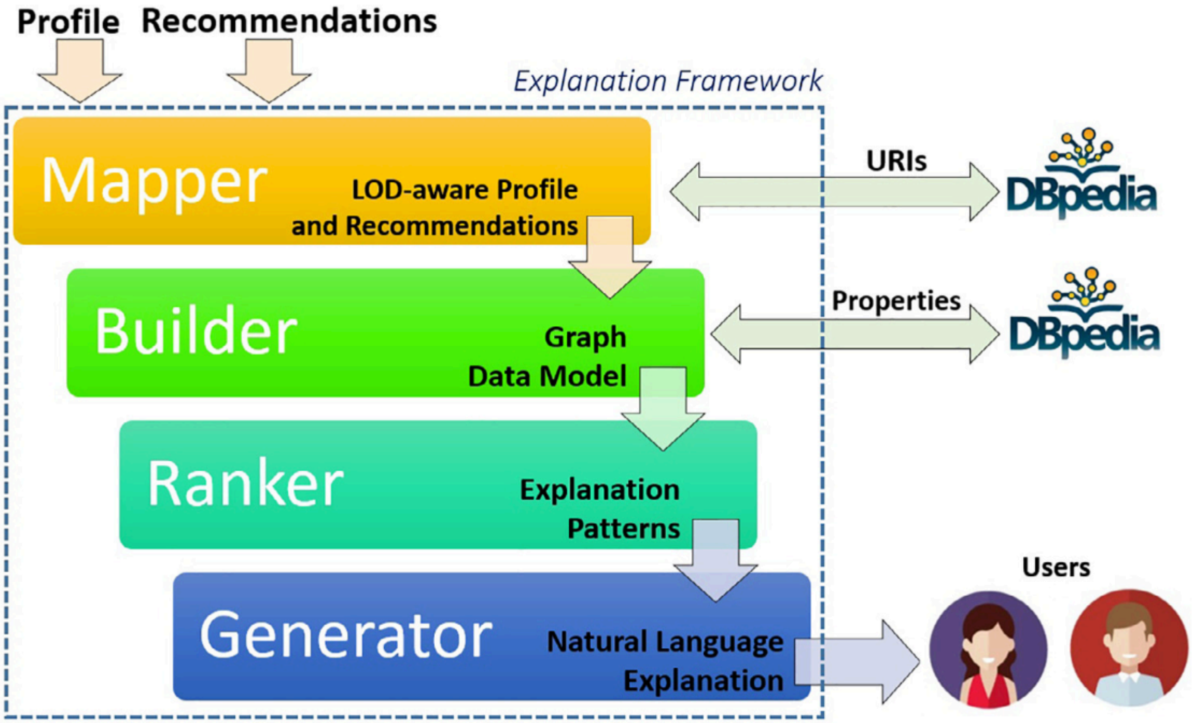
The Linked Open Data Cloud from lod-cloud.net

Name	Entities	Relations	Types	Facts
Freebase	40M	35K	26.5K	637M
DBpedia (en)	4.6M	1.4K	735	580M
YAGO3	17M	77	488K	150M
Wikidata	15.6M	1.7K	23.2K	66M

Knowledge Graphs and Explainable AI

We can use Knowledge Graphs for *explaining* the decisions of Machine Learning algorithms, such as recommender systems, and design machine learning models that are less prone to capturing *spurious correlations* in the data.

- Locally vs. Globally
- Ad-hoc vs. Post-hoc

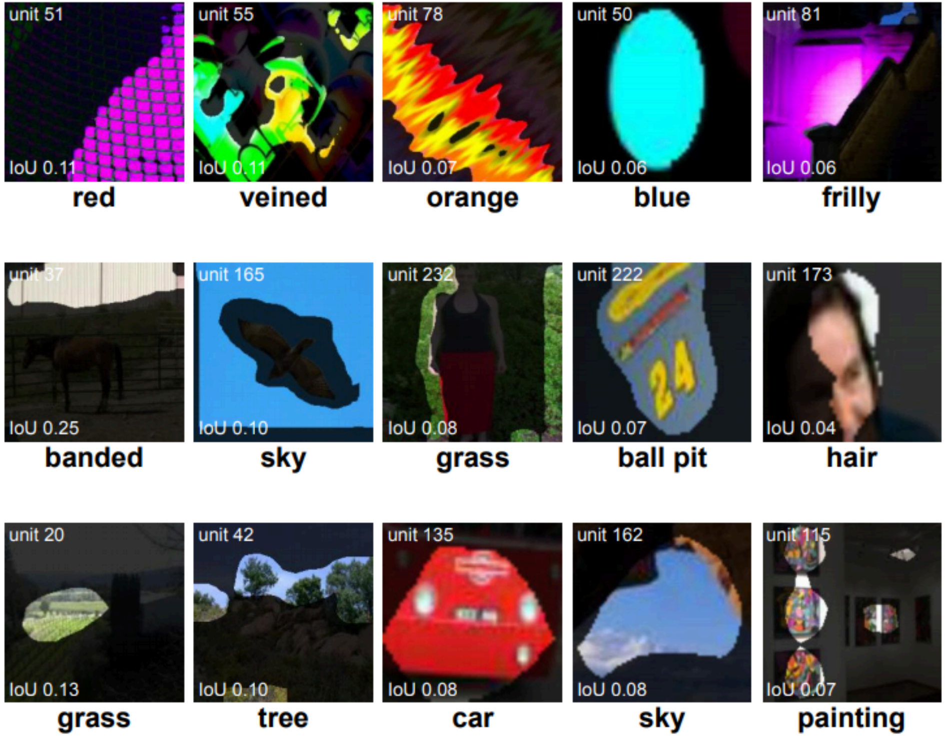


LOD-based Explanations for Transparent Recommender Systems - IJHCS
Linked Open Data to Support Content-Based Recommender Systems - ICSS
Top-n recommendations from implicit feedback leveraging linked open data - RECSYS

Knowledge Graphs and Explainable AI

We can use Knowledge Graphs for *explaining* the decisions of Machine Learning algorithms, such as recommender systems, and design machine learning models that are less prone to capturing *spurious correlations* in the data.

- Locally vs. Globally
- Ad-hoc vs. Post-hoc



Network Dissection: Quantifying Interpretability of Deep Visual Representations
On the Role of Knowledge Graphs in Explainable AI - SWJ

Knowledge Graphs and Explainable AI

We can use Knowledge Graphs for *explaining* the decisions of Machine Learning algorithms, such as recommender systems, and design machine learning models that are less prone to capturing *spurious correlations* in the data.

- Locally vs. Globally
- Ad-hoc vs. Post-hoc

Annotation Artifacts in Natural Language Inference Data

Suchin Gururangan[★◇] Swabha Swayamdipta[★♡]
Omer Levy[♣] Roy Schwartz[♣♠] Samuel R. Bowman[†] Noah A. Smith[♣]

Performance Impact Caused by Hidden Bias of Training Data for Recognizing Textual Entailment

Masatoshi Tsuchiya

Behavior Analysis of NLI Models: Uncovering the Influence of Three Factors on Robustness

V. Ivan Sanchez Carmona and Jeff Mitchell and Sebastian Riedel

Hypothesis Only Baselines in Natural Language Inference

Adam Poliak[1] Jason Naradowsky[1] Aparajita Haldar[1,2]
Rachel Rudinger[1] Benjamin Van Durme[1]

On the Role of Knowledge Graphs in Explainable AI - SWJ
Dynamic Integration of Background Knowledge in Neural NLU Systems

Knowledge Graphs Construction

Knowledge Graph construction methods can be classified in:

- **Manual** — <u>curated</u> (e.g. via experts), <u>collaborative</u> (e.g. via volunteers)
- **Automated** — <u>semi-structured</u> (e.g. from infoboxes), <u>unstructured</u> (e.g. from text)

Coverage is an issue:

- **Freebase** (40M entities) - 71% of persons without a birthplace, 75% without a nationality, even worse for other relation types [Dong et al. 2014]
- **DBpedia** (20M entities) - 61% of persons without a birthplace, 58% of scientists missing why they are popular [Krompaß et al. 2015]

Relational Learning can help us overcoming these issues and - in general - with learning from relational representations.

Relational Learning in Knowledge Graphs

- **Dyadic Multi-Relational Data** [Nickel et al. 2015, Getoor et al. 2007]

- Many possible relational learning tasks:
  - **Link Prediction** — Identify missing relationships between entities
  - **Collective Classification** — Classify entities based on their relationships
  - **Link-Based Clustering** — Cluster entities based on their relationships
  - **Entity Resolution** — Entity mapping/deduplication

Relational structure is a rich source of information.

In general, the *i.i.d. assumption* does not hold in this context.

Statistical Relational Learning

Task — model the existence of each triple $x_{spo} = (s, p, o) \in \mathscr{E} \times \mathscr{R} \times \mathscr{E}$ as *binary random variables* $y_{spo} \in \{0,1\}$ indicating whether x_{spo} is in the KG:

$$y_{spo} = \begin{cases} 1 & \text{if } x_{spo} \in \mathscr{G} \\ 0 & \text{otherwise} \end{cases} \quad \text{entries in} \quad \overline{\mathbf{Y}} \in \{0,1\}^{|\mathscr{E}|\times|\mathscr{R}|\times|\mathscr{E}|}$$

Every realisation of $\overline{\mathbf{Y}}$ denotes a *possible world* - modelling $P\left(\overline{\mathbf{Y}}\right)$ allows predicting triples based on the state of the entire Knowledge Graph.

Scalability is important - e.g. on Freebase (40M entities), the number of variables to represent can be quite large: $|\mathscr{E} \times \mathscr{R} \times \mathscr{E}| > 10^{19}$

Types of Statistical Relational Learning Models

Depending on our assumptions on $P\left(\overline{\mathbf{Y}}\right)$, we end up with *three model classes*:

- **Latent Feature Models**: variables $y_{spo} \in \{0,1\}$ are *conditionally independent* given the *latent features* $\mathbf{\Theta}$ associated with subject, predicate, and object:

$$\forall x_i, x_j \in \mathscr{E} \times \mathscr{R} \times \mathscr{E}, x_i \neq x_j : y_i \perp\!\!\!\perp y_j \mid \mathbf{\Theta}$$

- **Observable Feature Models**: related to Latent Feature Models, but $\mathbf{\Theta}$ are now *graph-based features*, such as *paths* linking the subject and the object.
- **Graphical Models**: variables $y_{spo} \in \{0,1\}$ are not assumed to be conditionally independent — each y_{spo} can depend on any of the other random variables in $\overline{\mathbf{Y}}$.

Conditional Independence Assumption

Assuming all y_{spo} variables are conditionally independent allows modelling their existence via a *scoring function* $f\left(s, p, o \mid \Theta\right)$ representing the likelihood that a triple is in the KG, conditioned on the parameters $\boldsymbol{\Theta}$:

$$P\left(\overline{\mathbf{Y}} \mid \Theta\right) = \prod_{s \in \mathscr{E}} \prod_{p \in \mathscr{R}} \prod_{o \in \mathscr{E}} \begin{cases} P\left(y_{spo} \mid \Theta\right) & \text{if } y_{spo} = 1 \\ 1 - P\left(y_{spo} \mid \Theta\right) & \text{otherwise} \end{cases} \quad \text{with } P\left(y_{spo} \mid \Theta\right) = \sigma\left(f(s, p, o \mid \Theta)\right)$$

Scoring Function - depending on the type of features used by $f\left(\,\cdot \mid \Theta\right)$ we have two families of models - *Observable* and *Latent Feature Models*.

Observable Feature Models

Uni-Relational Similarity Measures: based on *homophily* — similar entities are likely to be related — and *neighbourhood similarity.*

- **Local**: derive similarity between entities from their local neighbourhood
  (e.g. Common Neighbours, Adamic-Adar Index [Adamic et al. 2003], Preferential Attachment [Barabási et al. 1999], ..)

- **Global**: derive similarity between entities using the whole graph
  (e.g. Katz Index [Katz, 1953], Leicht-Holme-Newman Index [Leicht et al. 2006], PageRank [Brin et al. 1998], ..)

- **Quasi-Local**: trade-off between computational complexity and predictive accuracy
  (e.g. Local Katz Index [Liben-Nowell et al. 2007], Local Random Walks [Liu et al. 2010], ..)

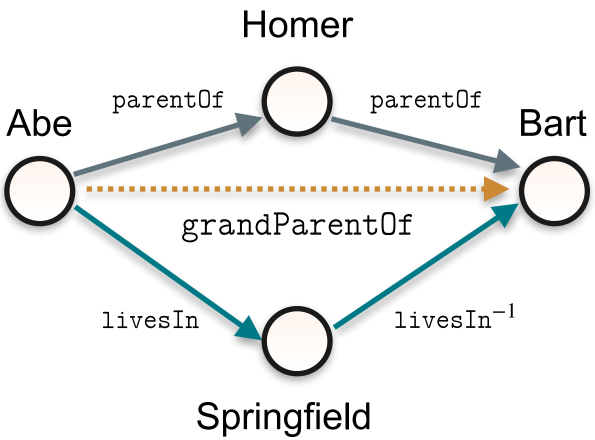
Observable Feature Models - Rule Mining and ILP

Rule Mining and **Inductive Logic Programming** methods extract rules via mining methods, and use them to infer new links.

- **Logic Programming (deductive):** from facts and rules, infer new facts (First-Order Logic)
- **Inductive Logic Programming (ILP):** from correlated facts, infer new rules
  (e.g. Progol [Muggleton, 1993], Aleph [Srinivasan, 1999], DL-Learner [Lehmann, 2009], FOIL [Quinlan, 1990], ..)
- **Rule Mining:** AMIE [Galárraga et al. 2015] is orders of magnitude faster than traditional ILP methods, and consistent with the Open World Assumption in Knowledge Graphs:
  - Partial Completeness Assumption
  - Efficient search space exploration via Mining Operators

Observable Feature Models - Path Ranking Algorithm

Path Ranking Algorithm (PRA) uses *length-bounded random walks* as features between entity pairs for predicting a target relation [Lao et al. 2010].



A **PRA model** scores a subject-object pair by a linear function of their path features:

$$f(s, p, o) = \sum_{\pi \in \Pi_p} P(s \rightarrow o \mid \pi) \times \theta_{\pi,p}$$

where Π is the set of all length-bounded relation paths, and θ are parameters estimated via L1,L2-regularised logistic regression.

Some extensions: Subgraph Features [Gardner et al. 2015], Multi-Task [Wang et al. 2016]

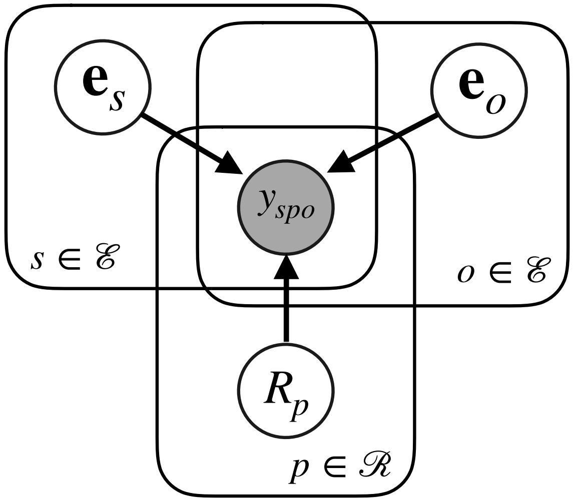
Observable Feature Models are *Interpretable*

Rules extracted by AMIE+ [Galárraga et al. 2015] from the YAGO3-10 dataset [Dettmers et al. 2018]

Body	$\Rightarrow$	Head	Confidence
$\texttt{hasNeighbor}(X, Y)$	$\Rightarrow$	$\texttt{hasNeighbor}(Y, X)$	0.99
$\texttt{isMarriedTo}(X, Y)$	$\Rightarrow$	$\texttt{isMarriedTo}(Y, X)$	0.96
$\texttt{hasNeighbor}(X, Z) \wedge \texttt{hasNeighbor}(Z, Y)$	$\Rightarrow$	$\texttt{hasNeighbor}(X, Y)$	0.88
$\texttt{isAffiliatedTo}(X, Y)$	$\Rightarrow$	$\texttt{playsFor}(Y, X)$	0.87
$\texttt{playsFor}(X, Y)$	$\Rightarrow$	$\texttt{isAffiliatedTo}(Y, X)$	0.75
$\texttt{dealsWith}(X, Z) \wedge \texttt{dealsWith}(Z, Y)$	$\Rightarrow$	$\texttt{dealsWith}(X, Y)$	0.73
$\texttt{isConnectedTo}(X, Y)$	$\Rightarrow$	$\texttt{isConnectedTo}(Y, X)$	0.66
$\texttt{dealsWith}(X, Z) \wedge \texttt{imports}(Z, Y)$	$\Rightarrow$	$\texttt{imports}(X, Y)$	0.61
$\texttt{influences}(Z, X) \wedge \texttt{isInterestedIn}(Z, Y)$	$\Rightarrow$	$\texttt{isInterestedIn}(X, Y)$	0.53

Latent Feature Models

Variables y_{spo} are conditionally independent given a set of latent features and parameters $\boldsymbol{\Theta}$. *Latent* means that are not directly observed in the data, and thus need to be estimated.

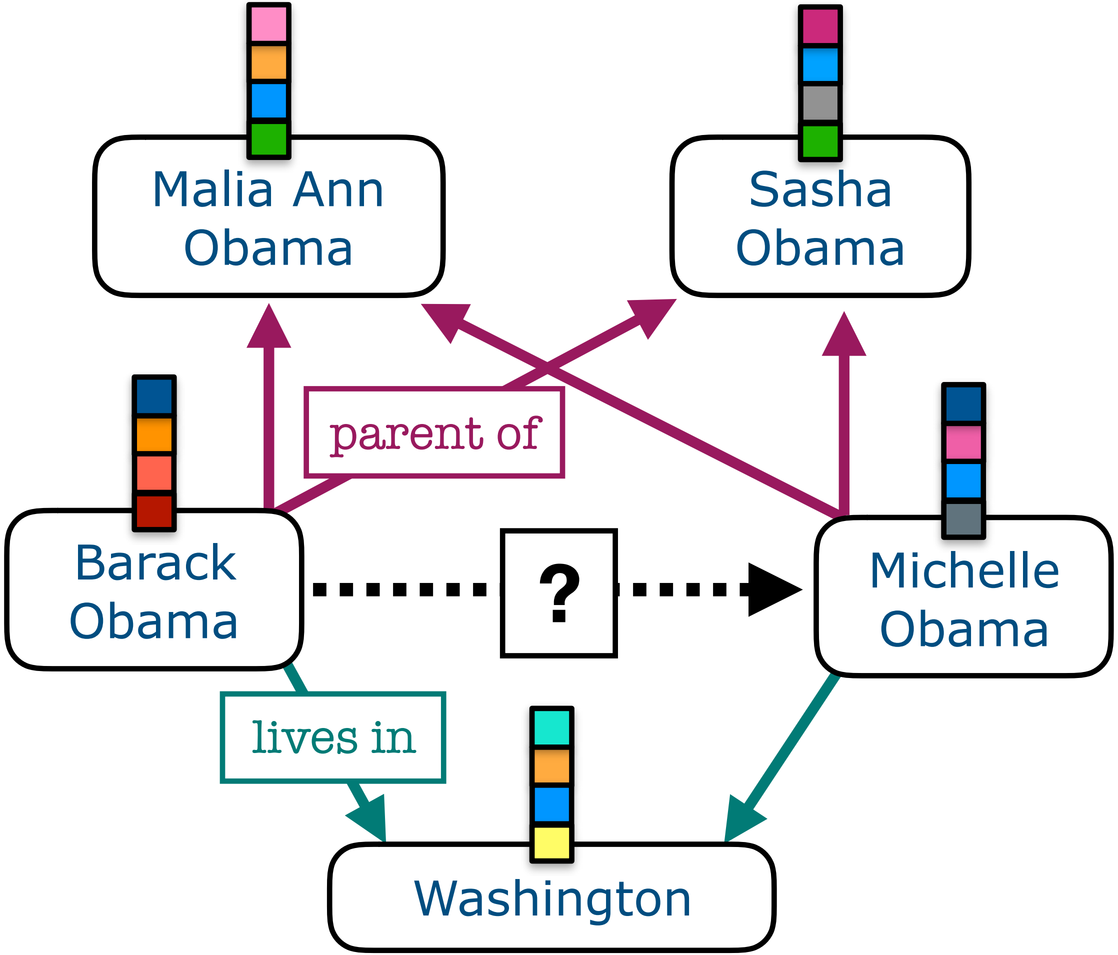


Relationships between entities *s* and *o* can be inferred from the interactions of their latent features $\mathbf{e}_s, \mathbf{e}_o$:

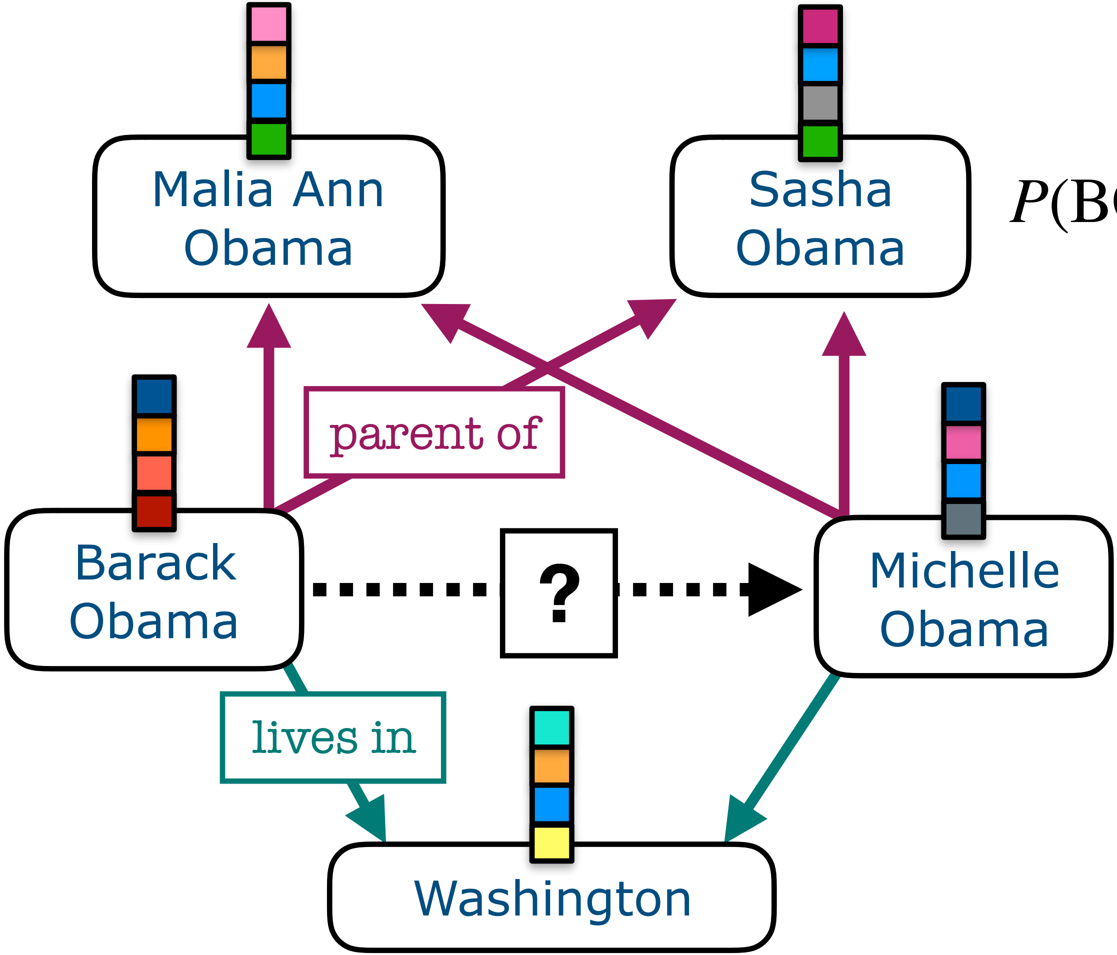
$$f(s, p, o) = f_p(\mathbf{e}_s, \mathbf{e}_o) \quad \begin{cases} \mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k, \\ f_p : \mathbb{R}^k \times \mathbb{R}^k \mapsto \mathbb{R} \end{cases}$$

The latent features inferred by these models can be <u>very hard to interpret</u>.

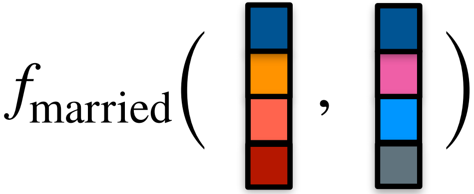
Latent Feature Models



Latent Feature Models



$$P(\mathrm{BO} \xrightarrow{\text{married}} \mathrm{MO}) \propto$$



Learning Representations

$$\mathcal{L}(\mathcal{G} \mid \Theta) = \sum_{(s,p,o)\in\mathcal{G}} \log \sigma\Big(f_p(\mathbf{e}_s, \mathbf{e}_o)\Big) + \sum_{(s,p,o)\notin\mathcal{G}} \log\Big[1 - \sigma\Big(f_p(\mathbf{e}_s, \mathbf{e}_o)\Big)\Big]$$

Latent Feature Models - Scoring Functions

Relationships between entities are determined by interactions between latent features — this yields different choices for the scoring function $f_p : \mathbb{R}^k \times \mathbb{R}^k \mapsto \mathbb{R}$:

Models	Scoring Functions	Parameters
RESCAL [Nickel et al. 2011]	$\mathbf{e}_s^\top \mathbf{W}_p \mathbf{e}_o$	$\mathbf{W}_p \in \mathbb{R}^{k \times k}$
NTN [Socher et al. 2013]	$\mathbf{u}_p^\top f\left(\mathbf{e}_s \mathbf{W}_p^{[1\ldots d]} + \mathbf{V}_p \begin{bmatrix}\mathbf{e}_s\\ \mathbf{e}_o\end{bmatrix} + \mathbf{b}_p\right)$	$\mathbf{W}_p \in \mathbb{R}^{k^2 \times d}, \mathbf{V}_p \in \mathbb{R}^{2k \times d}, \mathbf{b}_p, \mathbf{u}_p \in \mathbb{R}^k$
TransE [Bordes et al. 2013]	$-\left\lVert \mathbf{e}_s + \mathbf{r}_p - \mathbf{e}_o \right\rVert_{1,2}^2$	$\mathbf{r}_p \in \mathbb{R}^k$
DistMult [Yang et al. 2014]	$\langle \mathbf{e}_s, \mathbf{r}_p, \mathbf{e}_o \rangle$	$\mathbf{r}_p \in \mathbb{R}^k$
HolE [Nickel et al. 2016]	$\mathbf{r}_p^\top \left(\mathscr{F}^{-1}\left[\overline{\mathscr{F}[\mathbf{e}_s]} \odot \mathscr{F}[\mathbf{e}_o]\right]\right)$	$\mathbf{r}_p \in \mathbb{R}^k$
ComplEx [Trouillon et al. 2016]	$\mathrm{Re}\left(\langle \mathbf{e}_s, \mathbf{r}_p, \overline{\mathbf{e}}_o \rangle\right)$	$\mathbf{r}_p \in \mathbb{C}^k$
ConvE [Dettmers et al. 2017]	$f\left(\mathrm{vec}\left(f\left([\overline{\mathbf{e}_s}; \overline{\mathbf{r}_p}] * \omega\right)\right)\mathbf{W}\right)\mathbf{e}_o$	$\mathbf{r}_p \in \mathbb{R}^k, \mathbf{W} \in \mathbb{R}^{c \times k}$

Latent Feature Models - Scoring Functions

Relationships between entities are determined by interactions between latent features — this yields different choices for the scoring function $f_p : \mathbb{R}^k \times \mathbb{R}^k \mapsto \mathbb{R}$:

Models	Scoring Functions	Parameters
RESCAL [Nickel et al. 2011]	$\boxed{\mathbf{e}_s^\top \mathbf{W}_p \mathbf{e}_o}$	$\mathbf{W}_p \in \mathbb{R}^{k \times k}$
NTN [Socher et al. 2013]	$\mathbf{u}_p^\top f\left(\mathbf{e}_s \mathbf{W}_p^{[1\ldots d]} + \mathbf{V}_p \begin{bmatrix}\mathbf{e}_s \\ \mathbf{e}_o\end{bmatrix} + \mathbf{b}_p\right)$	$\mathbf{W}_p \in \mathbb{R}^{k^2 \times d}, \mathbf{V}_p \in \mathbb{R}^{2k \times d}, \mathbf{b}_p, \mathbf{u}_p \in \mathbb{R}^k$
TransE [Bordes et al. 2013]	$-\left\Vert \mathbf{e}_s + \mathbf{r}_p - \mathbf{e}_o \right\Vert_{1,2}^{2}$	$\mathbf{r}_p \in \mathbb{R}^k$
DistMult [Yang et al. 2015]	$\langle \mathbf{e}_s, \mathbf{r}_p, \mathbf{e}_o \rangle$	$\mathbf{r}_p \in \mathbb{R}^k$
HolE [Nickel et al. 2016]	$\mathbf{r}_p^\top \left(\mathcal{F}^{-1}\left[\overline{\mathcal{F}[\mathbf{e}_s]} \odot \mathcal{F}[\mathbf{e}_o]\right]\right)$	$\mathbf{r}_p \in \mathbb{R}^k$
ComplEx [Trouillon et al. 2016]	$\mathrm{Re}\left(\langle \mathbf{e}_s, \mathbf{r}_p, \overline{\mathbf{e}}_o \rangle\right)$	$\mathbf{r}_p \in \mathbb{C}^k$
ConvE [Dettmers et al. 2017]	$f\left(\mathrm{vec}\left(f\left(\left[\overline{\mathbf{e}_s}; \overline{\mathbf{r}_p}\right] * \omega\right)\right)\mathbf{W}\right)\mathbf{e}_o$	$\mathbf{r}_p \in \mathbb{R}^k, \mathbf{W} \in \mathbb{R}^{c \times k}$

Latent Feature Models - Scoring Functions

Relationships between entities are determined by interactions between latent features — this yields different choices for the scoring function $f_p : \mathbb{R}^k \times \mathbb{R}^k \mapsto \mathbb{R}$:

Models	Scoring Functions	Parameters
RESCAL [Nickel et al. 2011]	$\boxed{\mathbf{e}_s^\top \mathbf{W}_p \mathbf{e}_o}$	$\mathbf{W}_p \in \mathbb{R}^{k \times k}$
NTN [Socher et al. 2013]	$\mathbf{u}_p^\top f\left(\mathbf{e}_s \mathbf{W}_p^{[1\dots d]} + \mathbf{V}_p \begin{bmatrix} \mathbf{e}_s \\ \mathbf{e}_o \end{bmatrix} + \mathbf{b}_p\right)$	$\mathbf{W}_p \in \mathbb{R}^{k^2 \times d}, \mathbf{V}_p \in \mathbb{R}^{2k \times d}, \mathbf{b}_p, \mathbf{u}_p \in \mathbb{R}^k$
TransE [Bordes et al. 2013]	$- \left\| \mathbf{e}_s + \mathbf{r}_p - \mathbf{e}_o \right\|_{1,2}^{2}$	$\mathbf{r}_p \in \mathbb{R}^k$
DistMult [Yang et al. 2015]	$\boxed{\langle \mathbf{e}_s, \mathbf{r}_p, \mathbf{e}_o \rangle}$	$\mathbf{r}_p \in \mathbb{R}^k$
HolE [Nickel et al. 2016]	$\mathbf{r}_p^\top \left(\mathcal{F}^{-1}\left[\overline{\mathcal{F}[\mathbf{e}_s]} \odot \mathcal{F}[\mathbf{e}_o]\right]\right)$	$\mathbf{r}_p \in \mathbb{R}^k$
ComplEx [Trouillon et al. 2016]	$\mathrm{Re}\left(\langle \mathbf{e}_s, \mathbf{r}_p, \bar{\mathbf{e}}_o \rangle\right)$	$\mathbf{r}_p \in \mathbb{C}^k$
ConvE [Dettmers et al. 2017]	$f\left(\mathrm{vec}\left(f\left([\overline{\mathbf{e}_s}; \overline{\mathbf{r}_p}] * \omega\right)\right) \mathbf{W}\right) \mathbf{e}_o$	$\mathbf{r}_p \in \mathbb{R}^k, \mathbf{W} \in \mathbb{R}^{c \times k}$

Latent Feature Models - Scoring Functions

Relationships between entities are determined by interactions between latent features — this yields different choices for the scoring function $f_p : \mathbb{R}^k \times \mathbb{R}^k \mapsto \mathbb{R}$:

Models	Scoring Functions	Parameters
RESCAL [Nickel et al. 2011]	$\boxed{\mathbf{e}_s^\top \mathbf{W}_p \mathbf{e}_o}$	$\mathbf{W}_p \in \mathbb{R}^{k \times k}$
NTN [Socher et al. 2013]	$\mathbf{u}_p^\top f\left(\mathbf{e}_s \mathbf{W}_p^{[1\ldots d]} + \mathbf{V}_p \begin{bmatrix} \mathbf{e}_s \\ \mathbf{e}_o \end{bmatrix} + \mathbf{b}_p\right)$	$\mathbf{W}_p \in \mathbb{R}^{k^2 \times d}, \mathbf{V}_p \in \mathbb{R}^{2k \times d}, \mathbf{b}_p, \mathbf{u}_p \in \mathbb{R}^k$
TransE [Bordes et al. 2013]	$-\left\Vert \mathbf{e}_s + \mathbf{r}_p - \mathbf{e}_o \right\Vert_{1,2}^{2}$	$\mathbf{r}_p \in \mathbb{R}^k$
DistMult [Yang et al. 2015]	$\boxed{\langle \mathbf{e}_s, \mathbf{r}_p, \mathbf{e}_o \rangle}$	$\mathbf{r}_p \in \mathbb{R}^k$
HolE [Nickel et al. 2016]	$\mathbf{r}_p^\top \left(\mathcal{F}^{-1}\left[\overline{\mathcal{F}[\mathbf{e}_s]} \odot \mathcal{F}[\mathbf{e}_o]\right]\right)$	$\mathbf{r}_p \in \mathbb{R}^k$
ComplEx [Trouillon et al. 2016]	$\boxed{\mathrm{Re}\left(\langle \mathbf{e}_s, \mathbf{r}_p, \bar{\mathbf{e}}_o \rangle\right)}$	$\mathbf{r}_p \in \mathbb{C}^k$
ConvE [Dettmers et al. 2017]	$f\left(\mathrm{vec}\left(f\left([\overline{\mathbf{e}_s}; \overline{\mathbf{r}_p}] * \omega\right)\right)\mathbf{W}\right)\mathbf{e}_o$	$\mathbf{r}_p \in \mathbb{R}^k, \mathbf{W} \in \mathbb{R}^{c \times k}$

Latent Feature Models - Learning

Another core differente among models is the *loss function* minimised for fitting the latent parameters $\boldsymbol{\Theta}$ to the data — let $f_{spo} = f\left(x_{spo} \mid \Theta\right)$ and $p_{spo} = \sigma\left(f_{spo}\right)$:

Losses	Formulation	Models
Quadratic Loss	$\sum_{(x_{spo}, y_{spo}) \in \mathscr{D}} \left\| y_{spo} - f_{spo} \right\|_2^2$	Tensor Factorisation, RESCAL (ALS)
Pairwise Loss	$\sum_{x_+ \in \mathscr{D}_+} \sum_{x_- \in \mathscr{D}_-} \mathscr{L}(x_+, x_-) \overset{e.g.}{=} \max\left\{0, \gamma + f_{x_-} - f_{x_+}\right\}$	SE, NTN, TransE, HolE
Cross-Entropy Loss	$\sum_{(x,y) \in \mathscr{D}} \left[y \log\left(p_x\right) + (1 - y) \log\left(1 - p_x\right) \right]$	ComplEx
Multiclass Loss	$\sum_{x_{spo} \in \mathscr{D}_+} \mathscr{L}(p_{spo}, 1) + \sum_{\tilde{s} \in \mathscr{E}} \mathscr{L}(p_{\tilde{s}po}, y_{\tilde{s}po}) + \sum_{\tilde{o} \in \mathscr{E}} \mathscr{L}(p_{sp\tilde{o}}, y_{sp\tilde{o}})$	ConvE, ComplEx-N3 [Dettmers et al. 2017, Lacroix et al. 2018]

Latent Feature Models - Predictive Accuracy

Evaluation Metrics — Area Under the Precision-Recall Curve (AUC-PR), Mean Reciprocal Rank (MRR), Hits@k. In MRR and Hits@k, for each test triple:

- Modify its subject with all the entities in the Knowledge Graph,
- Score all the triple variants, and *compute the rank* of the original test triple,
- Repeat for the object.

$$\text{MRR} = \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \frac{1}{\text{rank}_i}, \quad \text{HITS}@k = \frac{|\{\text{rank}_i \leq 10\}|}{|\mathcal{T}|}$$

From [Lacroix et al. ICML 2018]

	Model	WN18		WN18RR		FB15K		FB15K-237		YAGO3-10	
		MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10
Reciprocal	CP-FRO	**0.95**	0.95	0.46	0.48	**0.86**	0.91	0.34	0.51	0.54	0.68
	CP-N3	**0.95**	0.96	0.47	0.54	**0.86**	0.91	0.36	0.54	0.57	**0.71**
	ComplEx-FRO	**0.95**	0.96	0.47	0.54	**0.86**	0.91	0.35	0.53	0.57	**0.71**
	ComplEx-N3	**0.95**	0.96	**0.48**	**0.57**	**0.86**	0.91	**0.37**	**0.56**	**0.58**	**0.71**

Latent Feature Models - Interpreting the Embeddings

Learned relation embeddings — using *ComplEx* with a *pairwise margin-based loss* — for WordNet (left), DBpedia, and YAGO (right) [Minervini et al. ECML 2017]

Predicates	Real Part					Imaginary Part				
hypernym	1.0	3.0	-3.1	2.5	-2.7	3.2	2.9	1.7	-3.0	-3.0
hyponym	1.0	3.1	-3.1	2.6	-2.7	-3.4	-2.8	-1.7	2.9	3.0
synset domain topic of	-3.1	-2.7	2.2	3.2	-2.4	-3.0	-1.6	-2.9	-2.8	2.6
member of domain topic	-3.1	-2.7	2.2	3.2	-2.5	2.8	1.7	2.9	2.9	-2.6
member of domain usage	-1.4	-0.1	-2.5	-3.4	2.7	-3.0	1.8	2.6	-0.6	-1.3
synset domain usage of	-1.2	-0.1	-2.3	-3.3	2.6	3.1	-1.8	-2.5	0.7	1.4
instance hypernym	-1.1	-2.8	1.6	2.7	-2.5	3.0	-2.6	2.6	-1.1	-2.8
instance hyponym	-1.0	-2.9	1.5	2.9	-2.4	-2.9	2.8	-2.6	1.1	2.8
part of	-2.4	3.2	2.7	-1.5	3.0	-2.4	-0.6	-2.6	2.9	-1.9
has part	-2.5	3.2	2.9	-1.5	3.0	2.4	0.7	2.8	-3.0	1.9
member holonym	2.4	2.8	2.4	1.9	-2.4	2.9	-2.3	2.6	2.7	-2.4
member meronym	2.4	2.9	2.4	1.9	-2.3	-2.9	2.3	-2.5	-2.8	2.5
synset domain region of	-3.1	-0.3	3.1	-3.3	1.9	-0.9	2.0	-2.1	-1.2	1.0
member of domain region	-3.1	-0.3	3.2	-3.4	2.0	1.0	-2.1	2.2	1.3	-1.1
verb group	3.5	3.4	3.3	-1.8	-2.8	0.0	-0.1	0.0	0.0	0.0
derivationally related form	3.5	3.4	-3.2	3.4	3.2	0.0	0.0	-0.0	0.0	0.0

Predicates	Real Part					Imaginary Part				
musical arist	1.9	3.8	3.8	-1.7	-1.0	-2.5	0.4	-0.8	3.0	3.7
musical band	1.8	3.8	4.1	-1.8	-1.0	-2.5	0.3	-0.9	3.1	3.6
associated musical arist	3.7	3.2	3.7	3.4	3.3	0.7	0.1	0.2	-1.5	1.5
associated band	3.7	3.7	3.2	3.7	3.6	0.7	0.0	0.2	-1.5	1.5

Predicates	Real Part					Imaginary Part				
playsFor	3.6	-2.6	2.6	2.7	-3.1	2.5	3.0	2.8	2.6	-2.6
isAffiliatedTo	3.8	-2.6	2.6	2.6	-3.2	2.7	3.3	3.0	2.6	-2.8
hasNeighbor	0.9	2.5	2.9	3.5	2.2	0.0	-0.0	0.0	-0.1	-0.0
isMarriedTo	3.9	3.5	4.3	-2.1	0.0	0.0	-0.0	-0.0	0.0	0.0
isConnectedTo	-0.7	3.0	2.6	0.3	2.7	0.3	-0.1	-0.0	0.1	-0.0

Latent Feature Models - Interpreting the Embeddings

Learned relation embeddings — using *ComplEx* with a *pairwise margin-based loss* — for WordNet (left), DBpedia, and YAGO (right) [Minervini et al. ECML 2017]

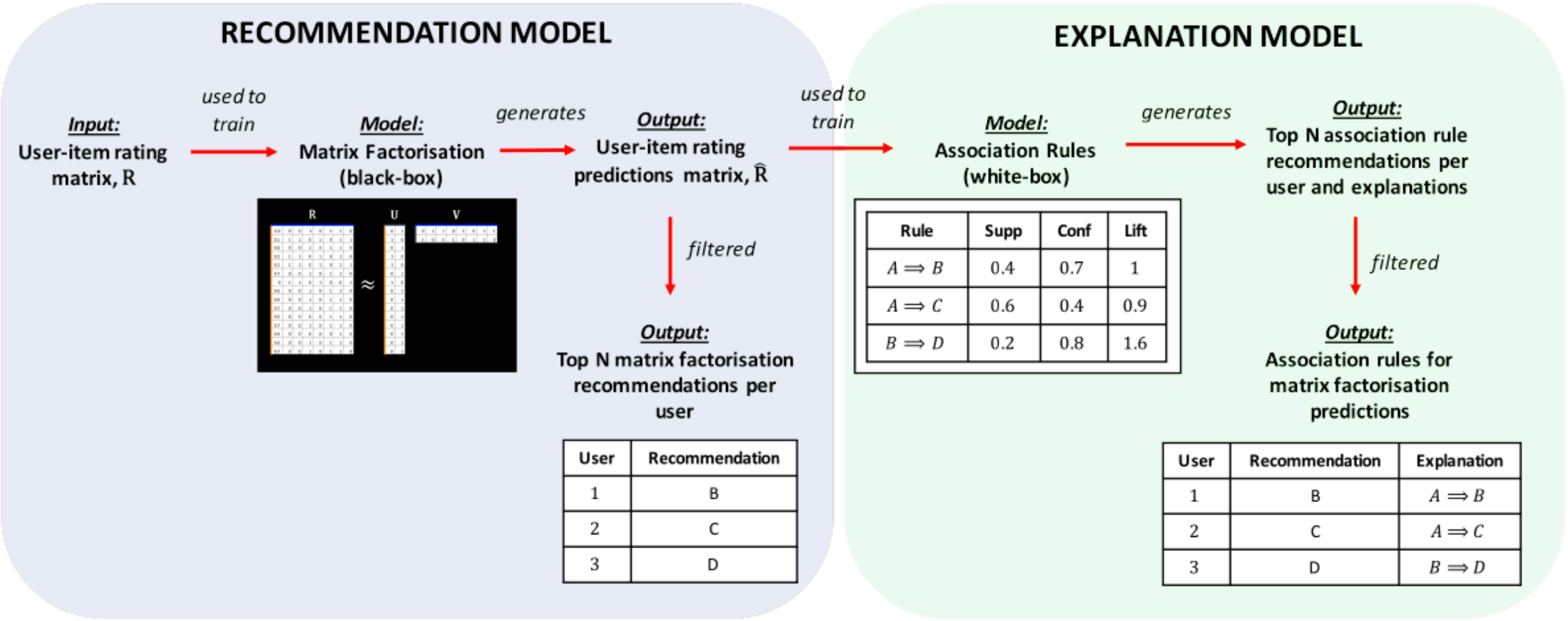
Predicates	Real Part					Imaginary Part				
hypernym	1.0	3.0	-3.1	2.5	-2.7	3.2	2.9	1.7	-3.0	-3.0
hyponym	1.0	3.1	-3.1	2.6	-2.7	-3.4	-2.8	-1.7	2.9	3.0
synset domain topic of	-3.1	-2.7	2.2	3.2	-2.4	-3.0	-1.6	-2.9	-2.8	2.6
member of domain topic	-3.1	-2.7	2.2	3.2	-2.5	2.8	1.7	2.9	2.9	-2.6
member of domain usage	-1.4	-0.1	-2.5	-3.4	2.7	-3.0	1.8	2.6	-0.6	-1.3
synset domain usage of	-1.2	-0.1	-2.3	-3.3	2.6	3.1	-1.8	-2.5	0.7	1.4
instance hypernym	-1.1	-2.8	1.6	2.7	-2.5	3.0	-2.6	2.6	-1.1	-2.8
instance hyponym	-1.0	-2.9	1.5	2.9	-2.4	-2.9	2.8	-2.6	1.1	2.8
part o	-2.4	3.2	2.7	-1.5	3.0	-2.4	-0.6	-2.6	2.9	-1.9
has pa	-2.5	3.2	2.9	-1.5	3.0	2.4	0.7	2.8	-3.0	1.9
member holonym	2.4	2.8	2.4	1.9	-2.4	2.9	-2.3	2.6	2.7	-2.4
member meronym	2.4	2.9	2.4	1.9	-2.3	-2.9	2.3	-2.5	-2.8	2.5
synset domain region of	-3.1	-0.3	3.1	-3.3	1.9	-0.9	2.0	-2.1	-1.2	1.0
member of domain region	-3.1	-0.3	3.2	-3.4	2.0	1.0	-2.1	2.2	1.3	-1.1
verb grou	3.5	3.4	3.3	-1.8	-2.8	0.0	-0.1	0.0	0.0	0.0
derivationally related for	3.5	3.4	-3.2	3.4	3.2	0.0	0.0	-0.0	0.0	0.0

Predicates	Real Part					Imaginary Part				
musical aris	1.9	3.8	3.8	-1.7	-1.0	-2.5	0.4	-0.8	3.0	3.7
musical ban	1.8	3.8	4.1	-1.8	-1.0	-2.5	0.3	-0.9	3.1	3.6
associated musical arist	3.7	3.2	3.7	3.4	3.3	0.7	0.1	0.2	-1.5	1.5
associated band	3.7	3.7	3.2	3.7	3.6	0.7	0.0	0.2	-1.5	1.5

Predicates	Real Part					Imaginary Part				
playsFor	3.6	-2.6	2.6	2.7	-3.1	2.5	3.0	2.8	2.6	-2.6
isAffiliatedTo	3.8	-2.6	2.6	2.6	-3.2	2.7	3.3	3.0	2.6	-2.8
hasNeighbor	0.9	2.5	2.9	3.5	2.2	0.0	-0.0	0.0	-0.1	-0.0
isMarriedTo	3.9	3.5	4.3	-2.1	0.0	0.0	-0.0	-0.0	0.0	0.0
isConnectedTo	-0.7	3.0	2.6	0.3	2.7	0.3	-0.1	-0.0	0.1	-0.0

Latent Feature Models - Post Hoc Interpretability

Generate an explanation model by training Bayesian Networks or Association Rules on the output of a Latent Feature Model. [Carmona et al. 2015, Peake et al. KDD 2018, Gusmão et al. 2018]



Combining Observable and Latent Feature Models

- **Additive Relational Effects (ARE)** [Nickel et al. NeurIPS 2014] — combines Observable and Latent Features in a single linear model:

$$f_{spo}^{ARE} = \mathbf{w}_{LFM,p}^{\top} \Theta_{LFM,so} + \mathbf{w}_{OBS,p}^{\top} \Theta_{PRA,so}$$

- **Knowledge Vault** [Dong et al. KDD 2014] — combines the prediction of Observable and Latent Feature Models via *stacking*:

$$f_{spo}^{KV} = f_{FUSION}\left(f_{spo}^{OFM}, f_{spo}^{LFM}\right)$$

- **Adversarial Sets** [Minervini et al. UAI 2017] — incorporate observable features, in the form of *First-Order Logic Rules R*, in Latent Feature Models:

$$\mathscr{L}(\Theta \mid R) = \mathscr{L}_{LFM}(\Theta) + \max_{\mathscr{S} \subseteq \mathscr{P}(\mathscr{E})} \mathscr{L}_{RULE}(\Theta, R)$$

End-to-End Differentiable Reasoning

We can combine *neural networks* and *symbolic models* by re-implementing classic reasoning algorithms using end-to-end differentiable (neural) architectures:

Differentiable Architectures

- Can generalise from high-dimensional, noisy, ambiguous inputs (*e.g.* sensory)
- Not interpretable
- Hard to incorporate knowledge
- Propositional fixation [McCarthy, 1988]

Logic Reasoning Based Models

- Can learn from small data
- Issues with high-dimensional, noisy, ambiguous inputs (*e.g.* images)
- Easy to *interpret*, and can provide *explanations* in the form of reasoning steps used to derive a conclusion

Reasoning in a Nutshell — Forward Chaining

Forward Chaining — start with a list of *facts*, and work forward from the *antecedent P* to the *consequent* Q iteratively.

$$q(X) \leftarrow p(X)$$

$$p(a)$$

$$p(b)$$

$$p(c)$$

$$\dots$$

Reasoning in a Nutshell — Forward Chaining

Forward Chaining — start with a list of *facts*, and work forward from the *antecedent P* to the *consequent* Q iteratively.

$$q(X) \leftarrow p(X)$$

$$\begin{array}{ccc} p(a) & & p(a), q(a) \\ p(b) & \dashrightarrow & p(b), q(b) \\ p(c) & & p(c), q(c) \\ \ldots & & \ldots \end{array}$$

Reasoning in a Nutshell — Backward Chaining

Backward Chaining — start with a list of *goals*, and work backwards from the *consequent* Q to the *antecedent P* to see if any data supports any of the consequents.

$$q(X) \leftarrow p(X)$$

$$p(a) \qquad q(a)?$$

$$p(b)$$

$$p(c)$$

$$\ldots$$

You can see backward chaining as a *query reformulation strategy.*

Reasoning in a Nutshell — Backward Chaining

Backward Chaining — start with a list of *goals*, and work backwards from the *consequent Q* to the *antecedent P* to see if any data supports any of the consequents.

$$q(X) \leftarrow p(X)$$

$$p(a)$$
$$p(b)$$
$$p(c)$$
$$\ldots$$

$$q(a)?$$

$$p(a)$$

You can see backward chaining as a *query reformulation strategy.*

Reasoning in a Nutshell — Backward Chaining

Backward Chaining — start with a list of *goals*, and work backwards from the *consequent Q* to the *antecedent P* to see if any data supports any of the consequents.

$$q(X) \leftarrow p(X)$$

$p(a)$

$p(b)$

$p(c)$

$\ldots$

$q(a)?$

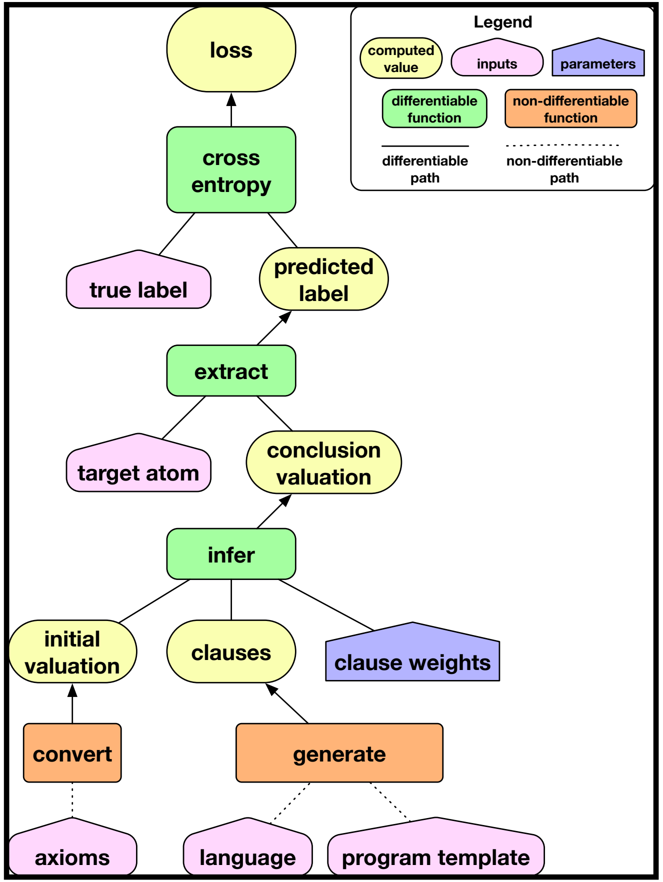
$\checkmark$

$p(a)$

You can see backward chaining as a *query reformulation strategy.*

Differentiable Forward Chaining - ∂ILP [Evans et al. JAIR 2018]

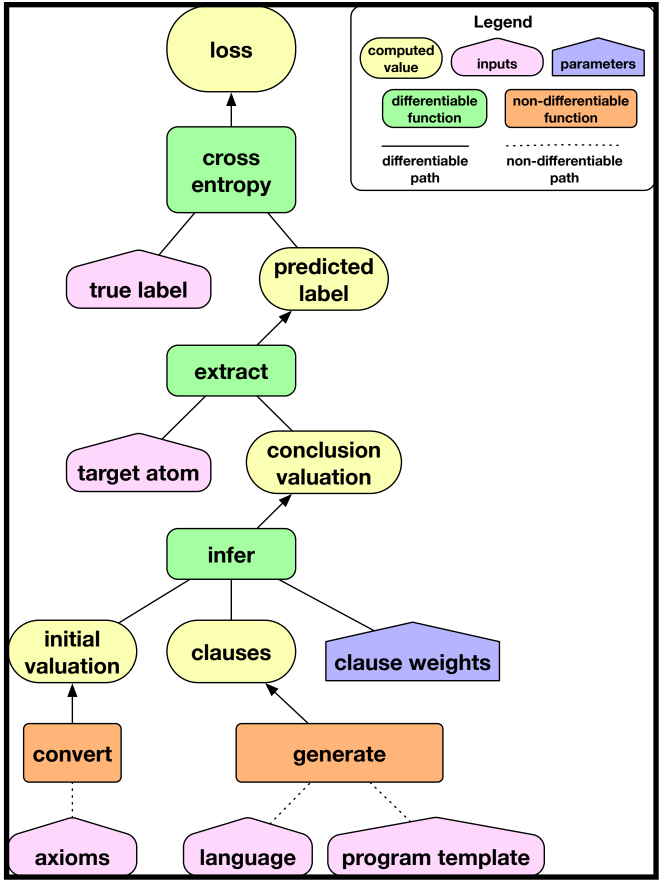
∂ILP uses a *differentiable model* of forward chaining inference:



Differentiable Forward Chaining - ∂ILP [Evans et al. JAIR 2018]

∂ILP uses a *differentiable model* of forward chaining inference:

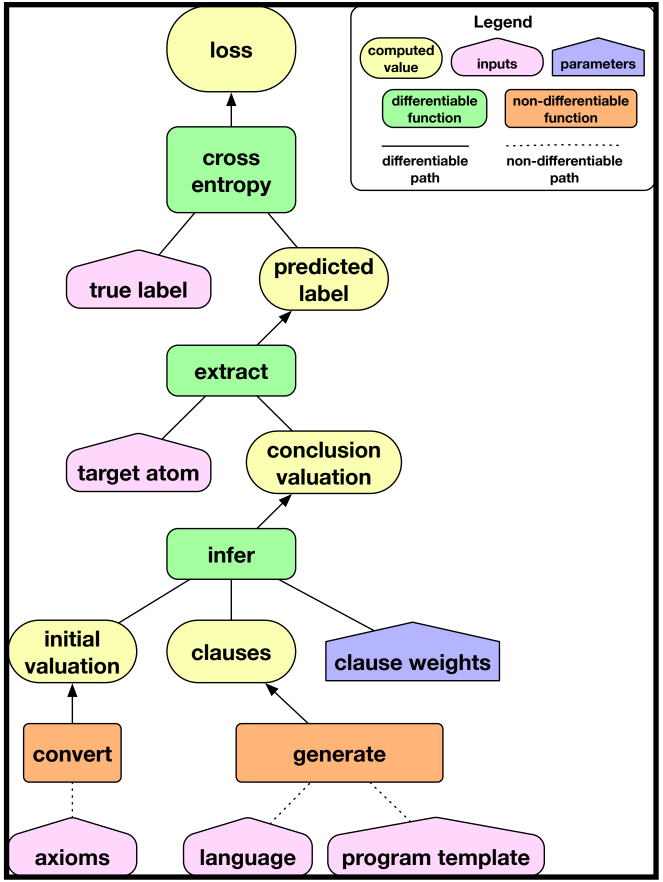
- Weights of the network represent a probability distribution over clauses



Differentiable Forward Chaining - ∂ILP [Evans et al. JAIR 2018]

∂ILP uses a *differentiable model* of forward chaining inference:

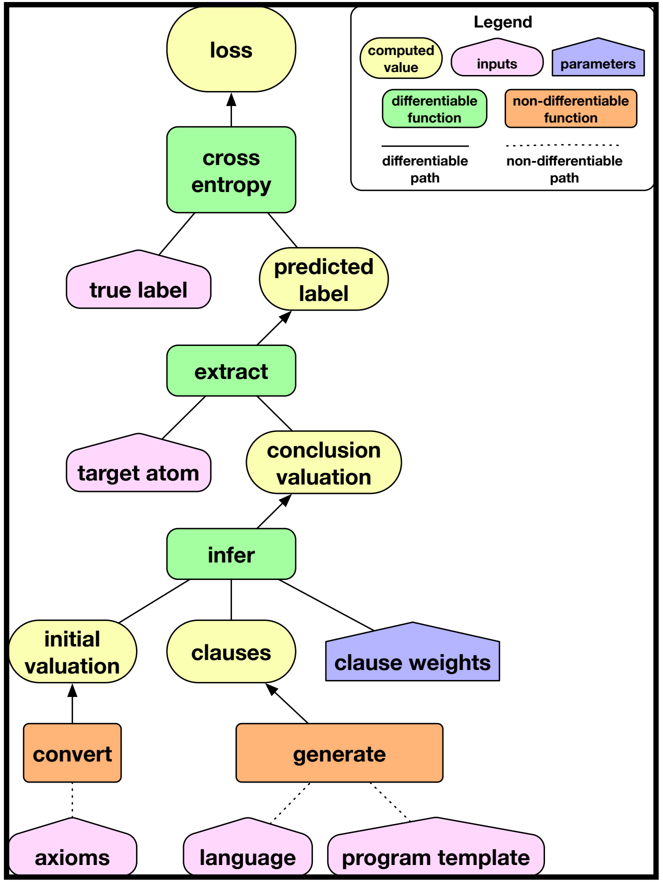
- Weights of the network represent a probability distribution over clauses
- A **valuation** is a vector with values in [0, 1] representing how likely it is that each of the **ground atoms** is true
- Forward chaining is implemented by a differentiable function that, given a valuation vector, produces another by applying **rules** to it.



Differentiable Forward Chaining - ∂ILP [Evans et al. JAIR 2018]

∂ILP uses a *differentiable model* of forward chaining inference:

- Weights of the network represent a probability distribution over clauses
- A **valuation** is a vector with values in [0, 1] representing how likely it is that each of the **ground atoms** is true
- Forward chaining is implemented by a differentiable function that, given a valuation vector, produces another by applying **rules** to it.
- If conclusions do not match the desired ones, the error is **back-propagated** to the weights.

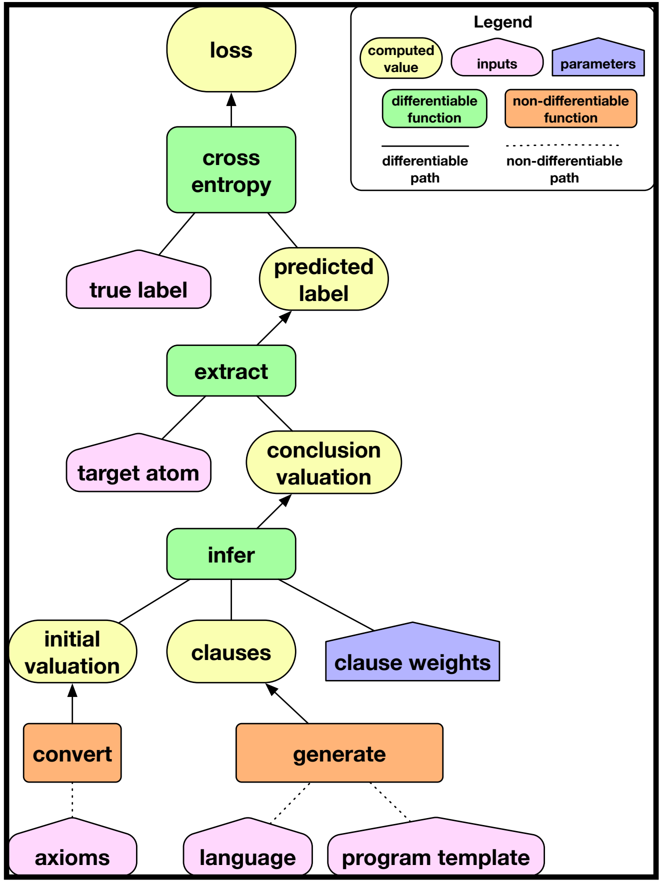


Differentiable Forward Chaining - ∂ILP [Evans et al. JAIR 2018]

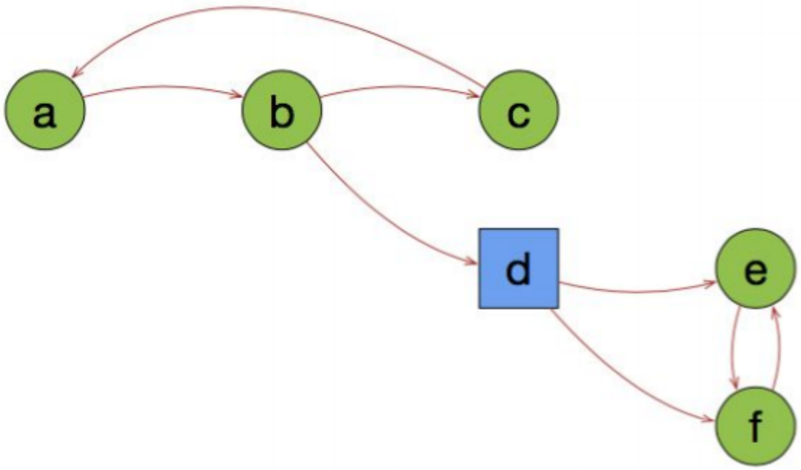
∂ILP uses a *differentiable model* of forward chaining inference:

- Weights of the network represent a probability distribution over clauses
- A **valuation** is a vector with values in [0, 1] representing how likely it is that each of the **ground atoms** is true
- Forward chaining is implemented by a differentiable function that, given a valuation vector, produces another by applying **rules** to it.
- If conclusions do not match the desired ones, the error is **back-propagated** to the weights.

We can extract a <u>readable program</u>.



Differentiable Forward Chaining - ∂ILP [Evans et al. JAIR 2018]



$$\text{cycle}(X) \leftarrow \text{pred}(X, X)$$
$$\text{pred}(X, Y) \leftarrow \text{edge}(X, Y)$$
$$\text{pred}(X, Y) \leftarrow \text{edge}(X, Z), \text{pred}(Z, Y)$$

Differentiable Forward Chaining - ∂ILP [Evans et al. JAIR 2018]

$$
\begin{aligned}
1 &\mapsto 1 \\
2 &\mapsto 2 \\
3 &\mapsto \mathit{Fizz} \\
4 &\mapsto 4 \\
5 &\mapsto \mathit{Buzz} \\
6 &\mapsto \mathit{Fizz} \\
7 &\mapsto 7 \\
8 &\mapsto 8 \\
9 &\mapsto \mathit{Fizz} \\
10 &\mapsto \mathit{Buzz}
\end{aligned}
$$

$$
\begin{aligned}
&\mathrm{fizz}(X) \leftarrow \mathrm{zero}(X) \\
&\mathrm{fizz}(X) \leftarrow \mathrm{fizz}(Y), \mathrm{pred1}(Y, X) \\
&\mathrm{pred1}(X, Y) \leftarrow \mathrm{succ}(X, Z), \mathrm{pred2}(Z, Y) \\
&\mathrm{pred2}(X, Y) \leftarrow \mathrm{succ}(X, Z), \mathrm{succ}(Z, Y)
\end{aligned}
$$

Backward Chaining — Differentiable Proving

[Rocktäschel et al. 2017, Minervini et al. 2018, Welbl et al. 2019]

Backward Chaining

$$q(X) \leftarrow p(X)$$

$p(a)$

$p(b)$

$p(c)$

$\dots$

$q(a)?$

$\checkmark$

$p(a)$

Backward Chaining — Differentiable Proving

[Rocktäschel et al. 2017, Minervini et al. 2018, Welbl et al. 2019]

Backward Chaining

$$q(X) \leftarrow p(X)$$

$p(a)$

$p(b)$

$p(c)$

$\dots$

$q(a)?$

✓

$p(a)$

BUT there's a problem..

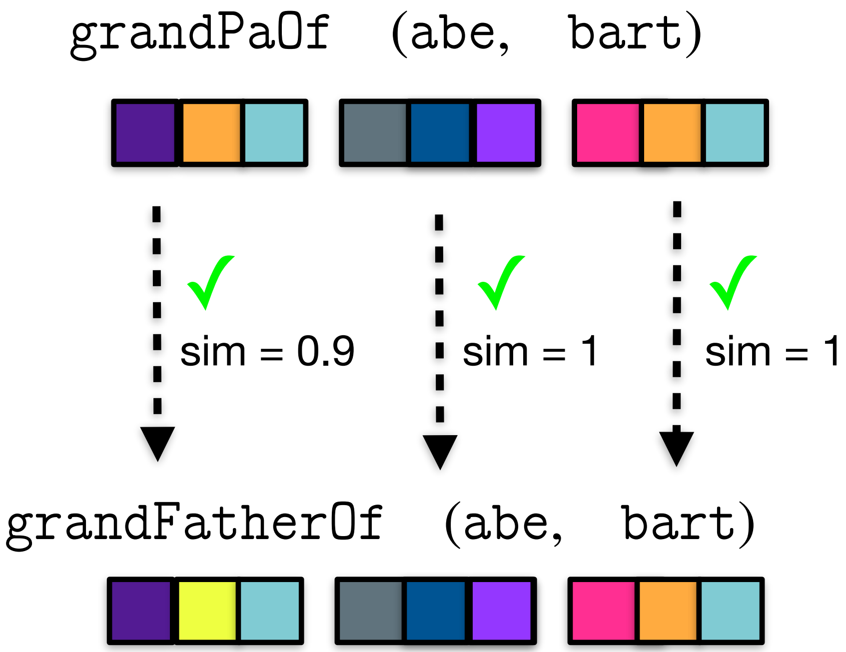
`grandPaOf (abe, bart)`

× ✓ ✓

`grandFatherOf (abe, bart)`

Backward Chaining — Differentiable Proving

[Rocktäschel et al. 2017, Minervini et al. 2018, Welbl et al. 2019]



Backward Chaining — Differentiable Proving

[Rocktäschel et al. 2017, Minervini et al. 2018, Welbl et al. 2019]

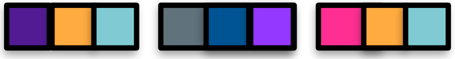
Knowledge Base:

$$\texttt{fatherOf}(\texttt{abe}, \texttt{homer})$$

$$\texttt{parentOf}(\texttt{homer}, \texttt{bart})$$

$$\texttt{grandFatherOf}(X, Y) \Leftarrow$$
$$\texttt{fatherOf}(X, Z),$$
$$\texttt{parentOf}(Z, Y).$$

$$\texttt{grandPaOf}(\texttt{abe}, \texttt{bart})$$



Backward Chaining — Differentiable Proving

[Rocktäschel et al. 2017, Minervini et al. 2018, Welbl et al. 2019]

Knowledge Base:

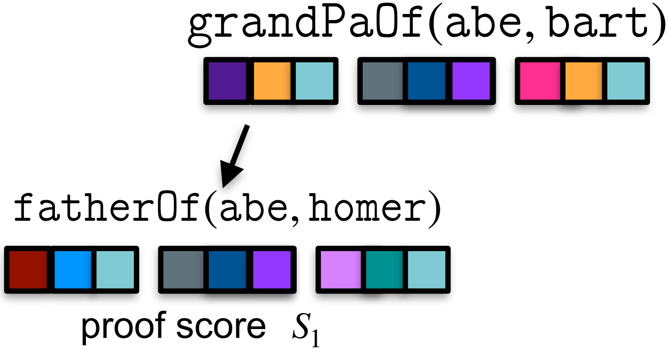
$\texttt{fatherOf}(\texttt{abe}, \texttt{homer})$

$\texttt{parentOf}(\texttt{homer}, \texttt{bart})$

$\texttt{grandFatherOf}(X, Y) \Leftarrow$

$\texttt{fatherOf}(X, Z),$

$\texttt{parentOf}(Z, Y).$



Backward Chaining — Differentiable Proving

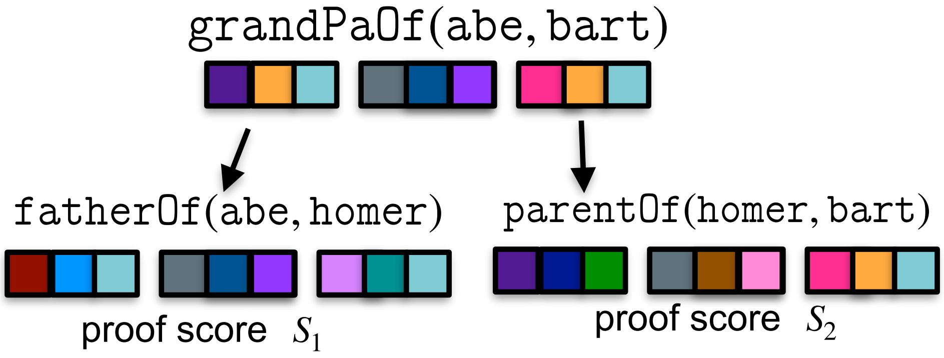
[Rocktäschel et al. 2017, Minervini et al. 2018, Welbl et al. 2019]

Knowledge Base:

$\texttt{fatherOf}(\texttt{abe}, \texttt{homer})$

$\texttt{parentOf}(\texttt{homer}, \texttt{bart})$

$\texttt{grandFatherOf}(X, Y) \Leftarrow$
$\texttt{fatherOf}(X, Z),$
$\texttt{parentOf}(Z, Y).$



Backward Chaining — Differentiable Proving

[Rocktäschel et al. 2017, Minervini et al. 2018, Welbl et al. 2019]

Knowledge Base:

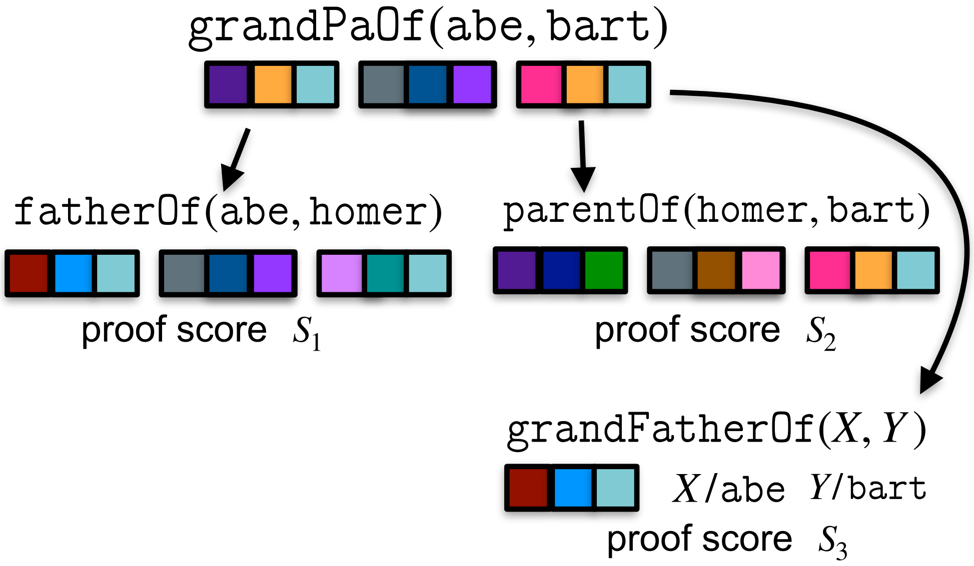
$\texttt{fatherOf(abe, homer)}$

$\texttt{parentOf(homer, bart)}$

$\texttt{grandFatherOf}(X, Y) \Leftarrow$

$\texttt{fatherOf}(X, Z),$

$\texttt{parentOf}(Z, Y).$



Subgoals:

$\texttt{fatherOf(abe,}\ Z\texttt{)}$

$\texttt{parentOf(}Z\texttt{, bart)}$

Backward Chaining — Differentiable Proving

[Rocktäschel et al. 2017, Minervini et al. 2018, Welbl et al. 2019]

Knowledge Base:

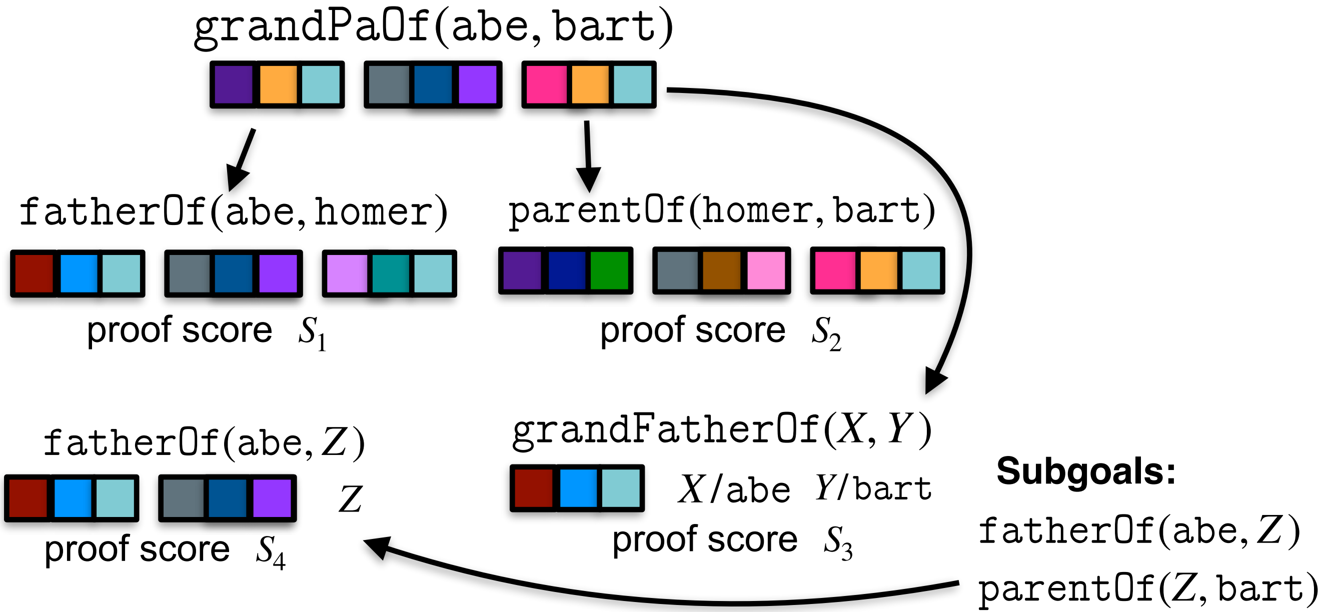
$\texttt{fatherOf(abe, homer)}$

$\texttt{parentOf(homer, bart)}$

$\texttt{grandFatherOf}(X, Y) \Leftarrow$

$\texttt{fatherOf}(X, Z),$

$\texttt{parentOf}(Z, Y).$



Backward Chaining — Differentiable Proving

[Rocktäschel et al. 2017, Minervini et al. 2018, Welbl et al. 2019]

Knowledge Base:

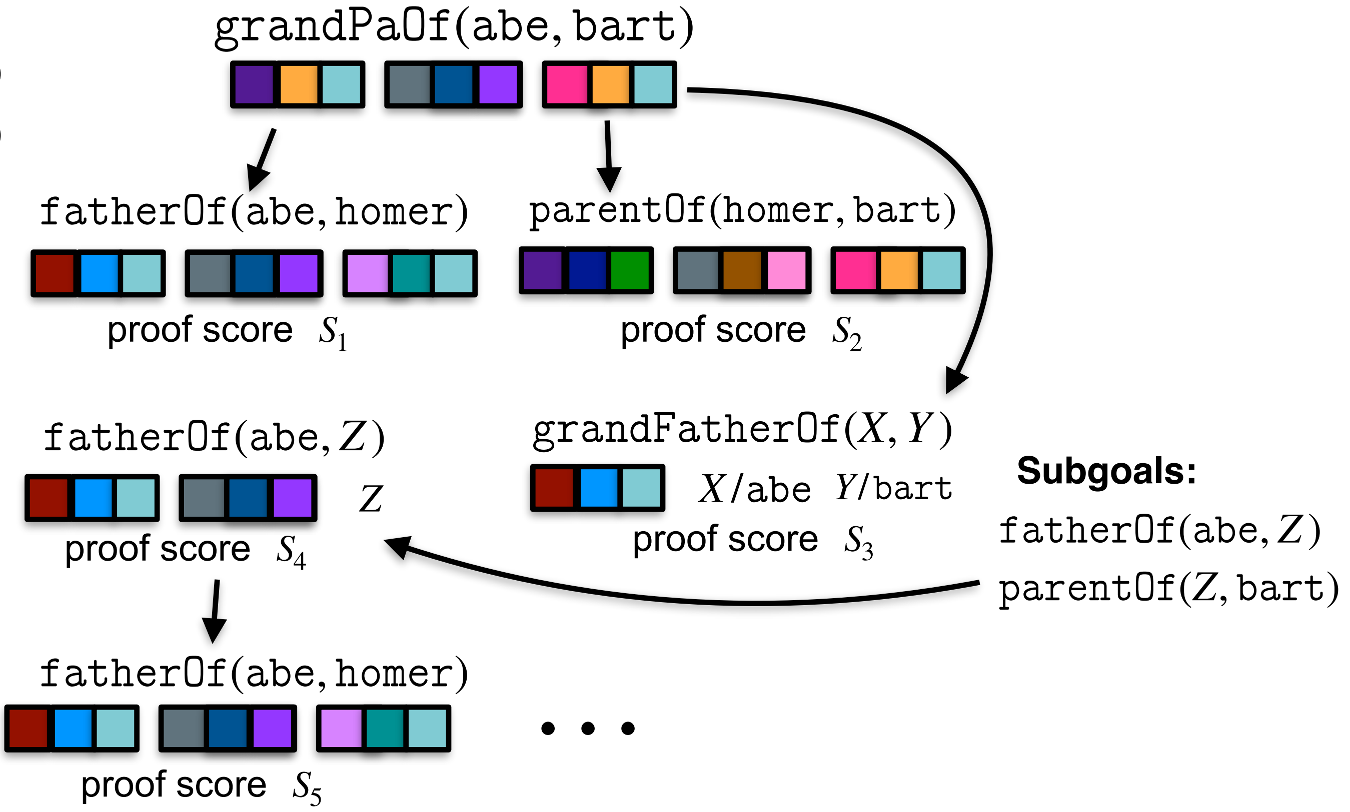
$\texttt{fatherOf}(\texttt{abe}, \texttt{homer})$

$\texttt{parentOf}(\texttt{homer}, \texttt{bart})$

$\texttt{grandFatherOf}(X, Y) \Leftarrow$

$\texttt{fatherOf}(X, Z),$

$\texttt{parentOf}(Z, Y).$



Learning Interpretable Rules From Data

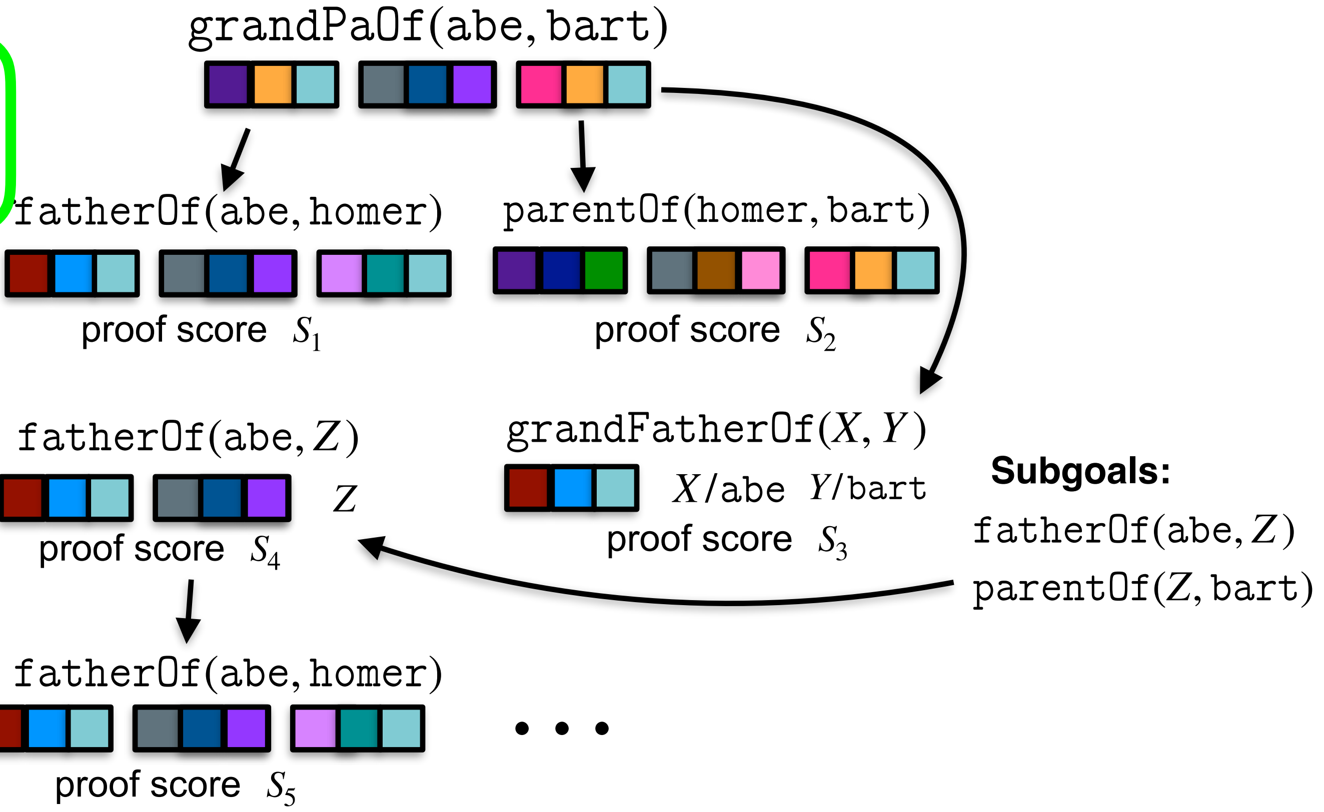
[Rocktäschel et al. 2017, Minervini et al. 2018, Welbl et al. 2019]

Knowledge Base:

`fatherOf(abe, homer)`
`parentOf(homer, bart)`
$\theta_1(X, Y) \Leftarrow \theta_2(X, Z), \theta_3(Z, Y)$

Training
Maximise Log-Likelihood:

$$\sum_{F \in K} \log p^{KB \setminus F}(F) - \sum_{\tilde{F} \sim corr(F)} \log p^{KB}(\tilde{F})$$



Differentiable Reasoning

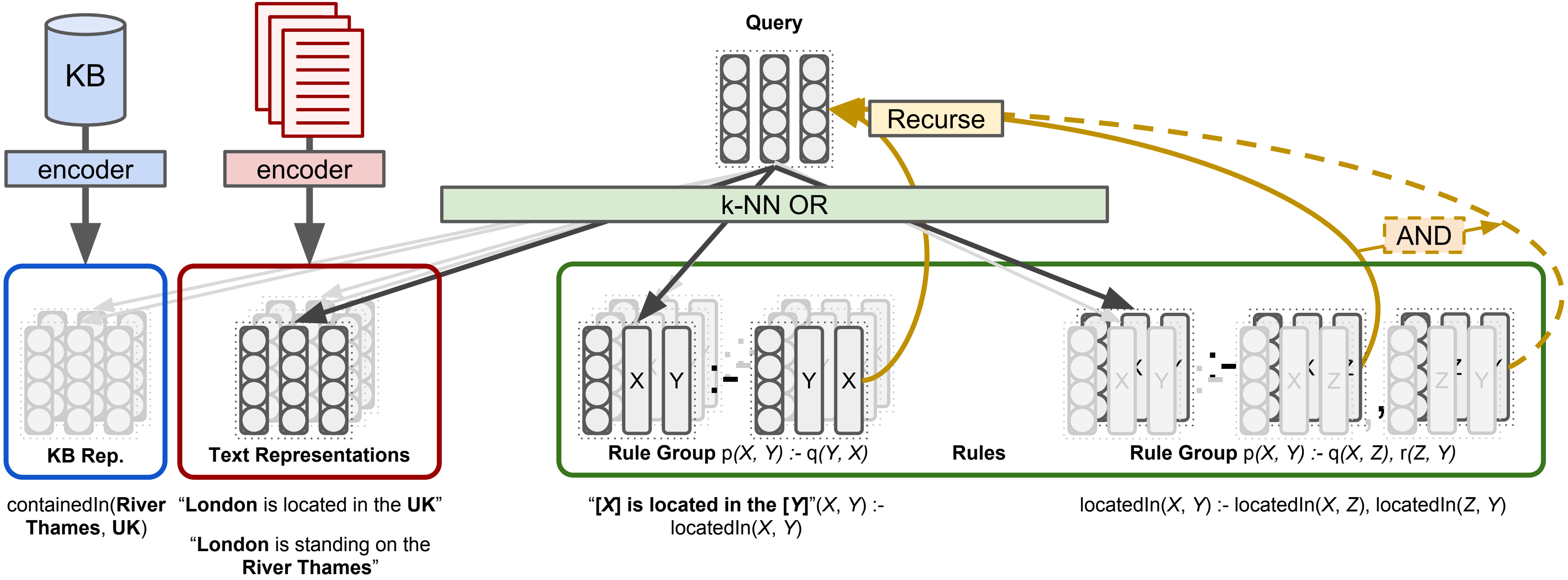
Corpus		Metric	Model			Examples of induced rules and their confidence
			ComplEx	NTP	NTPλ	
Countries	S1	AUC-PR	99.37 ± 0.4	90.83 ± 15.4	$\mathbf{100.00} \pm 0.0$	0.90 `locatedIn(X,Y) :– locatedIn(X,Z), locatedIn(Z,Y).`
	S2	AUC-PR	87.95 ± 2.8	87.40 ± 11.7	$\mathbf{93.04} \pm 0.4$	0.63 `locatedIn(X,Y) :– neighborOf(X,Z), locatedIn(Z,Y).`
	S3	AUC-PR	48.44 ± 6.3	56.68 ± 17.6	$\mathbf{77.26} \pm 17.0$	0.32 `locatedIn(X,Y) :– neighborOf(X,Z), neighborOf(Z,W), locatedIn(W,Y).`
Kinship		MRR	**0.81**	0.60	0.80	0.98 `term15(X,Y) :– term5(Y,X)`
		HITS@1	0.70	0.48	**0.76**	0.97 `term18(X,Y) :– term18(Y,X)`
		HITS@3	**0.89**	0.70	0.82	0.86 `term4(X,Y) :– term4(Y,X)`
		HITS@10	**0.98**	0.78	0.89	0.73 `term12(X,Y) :– term10(X, Z), term12(Z, Y).`
Nations		MRR	**0.75**	**0.75**	0.74	0.68 `blockpositionindex(X,Y) :– blockpositionindex(Y,X).`
		HITS@1	**0.62**	**0.62**	0.59	0.46 `expeldiplomats(X,Y) :– negativebehavior(X,Y).`
		HITS@3	0.84	0.86	**0.89**	0.38 `negativecomm(X,Y) :– commonbloc0(X,Y).`
		HITS@10	**0.99**	**0.99**	**0.99**	0.38 `intergovorgs3(X,Y) :– intergovorgs(Y,X).`
UMLS		MRR	0.89	0.88	**0.93**	0.88 `interacts_with(X,Y) :– interacts_with(X,Z), interacts_with(Z,Y).`
		HITS@1	0.82	0.82	**0.87**	0.77 `isa(X,Y) :– isa(X,Z), isa(Z,Y).`
		HITS@3	0.96	0.92	**0.98**	0.71 `derivative_of(X,Y) :– derivative_of(X,Z), derivative_of(Z,Y).`
		HITS@10	**1.00**	0.97	**1.00**	

Explainable Neural Link Prediction

	Query	Score S_ρ	Proofs / Explanations
WN18	part_of(CONGO.N.03, AFRICA.N.01)	0.995	part_of(X, Y) :– has_part(Y, X) has_part(AFRICA.N.01, CONGO.N.03)
		0.787	part_of(X, Y) :– instance_hyponym(Y, X) instance_hyponym(AFRICAN_COUNTRY.N.01, CONGO.N.03)
	hyponym(EXTINGUISH.V.04, DECOUPLE.V.03)	0.987	hyponym(X, Y) :– hypernym(Y, X) hypernym(DECOUPLE.V.03, EXTINGUISH.V.04)
		0.920	hypernym(SNUFF_OUT.V.01, EXTINGUISH.V.04)
	part_of(PITUITARY.N.01, DIENCEPHALON.N.01)	0.995	has_part(DIENCEPHALON.N.01, PITUITARY.N.01)
	has_part(TEXAS.N.01, ODESSA.N.02)	0.961	has_part(X, Y) :– part_of(Y, X) part_of(ODESSA.N.02, TEXAS.N.01)
	hyponym(SKELETAL_MUSCLE, ARTICULAR_MUSCLE)	0.987	hypernym(ARTICULAR_MUSCLE, SKELETAL_MUSCLE)
	deriv_related_form(REWRITE, REWRITING)	0.809	deriv_related_form(X, Y) :– hypernym(Y, X) hypernym(REVISE, REWRITE)
WN18RR	also_see(TRUE.A.01, FAITHFUL.A.01)	0.962	also_see(X, Y) :– also_see(Y, X) also_see(FAITHFUL.A.01, TRUE.A.01)
		0.590	also_see(CONSTANT.A.02, FAITHFUL.A.01)
	also_see(GOOD.A.03, VIRTUOUS.A.01)	0.962	also_see(VIRTUOUS.A.01, GOOD.A.03)
		0.702	also_see(RIGHTEOUS.A.01, VIRTUOUS.A.01)
	instance_hypernym(CHAPLIN, FILM_MAKER)	0.812	instance_hypernym(CHAPLIN, COMEDIAN)

Reasoning Over Text

We can embed facts from the KG and facts from text in a *shared embedding space*, and learn to reason over them *jointly:*



Reasoning Over Text

[Rocktäschel et al. 2017, Minervini et al. 2018, Welbl et al. 2019]

We can embed facts from the KG and facts from text in a *shared embedding space*, and learn to reason over them *jointly:*

Control Myself record_label Jam Recordings

$\text{record\_label}(X, Z) \leftarrow p_1(X, Y)$

$p_1(X, Z) \leftarrow p_2(X, Y) \land p_3(Y, Z)$

Control Myself [...] is a song by american rapper [...] Ell

Ell cools 1989 album [...] was released by [...] Jam Recordings

Reasoning Over Text

[Rocktäschel et al. 2017, Minervini et al. 2018, Welbl et al. 2019]

We can embed facts from the KG and facts from text in a *shared embedding space*, and learn to reason over them *jointly:*

Thrasyvoulos F.C. country Greece

$\text{country}(X, Z) \leftarrow p_1(X, Y)$

$p_1(X, Z) \leftarrow p_2(X, Y) \wedge p_3(Y, Z)$

Thrasyvoulos Fylis is a football club based in Fyli, Attica [...]

Fyli is a town and a municipality in the northwestern part of Attica, Greece

Neuro-Symbolic Integration — Recent Advances

- Recursive Reasoning Networks [Hohenecker et al. 2018] — given a OWL RL ontology, uses a differentiable model to update the entity and predicate representations.
- Deep ProbLog [Manhaeve et al. NeurIPS 2018] — extends the ProbLog probabilistic logic programming language with *neural predicates* that can be evaluated on e.g. sensory data (images, speech).
- Logic Tensor Networks [Serafini et al. 2016, 2017] — fully ground First Order Logic rules.
- AutoEncoder-like Architectures [Campero et al. 2018] — use end-to-end differentiable reasoning in the decoder of an autoencoder-like architecture to learn the minimal set of facts and rules that govern your domain via backprop.

Bibliography

Maximilian Nickel, Kevin Murphy, Volker Tresp, Evgeniy Gabrilovich:
A Review of Relational Machine Learning for Knowledge Graphs. Proceedings of the IEEE 104(1): 11-33 (2016)

Lise Getoor and Ben Taskar:
Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning). The MIT Press (2007)

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, Wei Zhang:
Knowledge vault: a web-scale approach to probabilistic knowledge fusion. KDD 2014: 601-610

Denis Krompaß, Stephan Baier, Volker Tresp:
Type-Constrained Representation Learning in Knowledge Graphs. International Semantic Web Conference (1) 2015: 640-655

L. A. Adamic and E. Adar:
Friends and neighbors on the Web. Social Networks, vol. 25, no. 3, pp. 211–230, 2003

A.-L. Barabási and R. Albert:
Emergence of Scaling in Random Networks. Science, vol. 286, no. 5439, pp. 509–512, 1999

L. Katz:
A new status index derived from sociometric analysis. Psychometrika, vol. 18, no. 1, pp. 39–43, 1953

E. A. Leicht, P. Holme, and M. E. Newman:
Vertex similarity in networks. Physical Review E, vol. 73, no. 2, p. 026120, 2006

S. Brin and L. Page:
The anatomy of a large-scale hypertextual Web search engine. Computer networks and ISDN systems, vol. 30, no. 1, pp. 107–117, 1998.

D. Liben-Nowell and J. Kleinberg:
The link-prediction problem for social networks. Journal of the American society for information science and technology, vol. 58, no. 7, pp. 1019–1031, 2007.

Bibliography

W. Liu and L. Lü:
Link prediction based on local random walk. EPL (Europhysics Letters), vol. 89, no. 5, p. 58007, 2010.

Stephen Muggleton:
Inverting Entailment and Progol. Machine Intelligence 14 1993: 135-190

Ashwin Srinivasan:
The Aleph Manual. http://www.di.ubi.pt/~jpaulo/competence/tutorials/aleph.pdf 1999

Jens Lehmann:
DL-Learner: Learning Concepts in Description Logics. Journal of Machine Learning Research 10: 2639-2642 (2009)

J. R. Quinlan:
Learning logical definitions from relations. Machine Learning, vol. 5, pp. 239–266, 1990

Ni Lao, Tom M. Mitchell, William W. Cohen:
Random Walk Inference and Learning in A Large Scale Knowledge Base. EMNLP 2011: 529-539

Luis Galárraga, Christina Teflioudi, Katja Hose, Fabian M. Suchanek:
Fast rule mining in ontological knowledge bases with AMIE+. VLDB J. 24(6): 707-730 (2015)

Maximilian Nickel, Volker Tresp, Hans-Peter Kriegel:
A Three-Way Model for Collective Learning on Multi-Relational Data. ICML 2011: 809-816

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, Oksana Yakhnenko:
Translating Embeddings for Modeling Multi-relational Data. NIPS 2013: 2787-2795

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, Li Deng:
Embedding Entities and Relations for Learning and Inference in Knowledge Bases. CoRR abs/1412.6575 (2014)

Bibliography

Maximilian Nickel, Lorenzo Rosasco, Tomaso A. Poggio:
Holographic Embeddings of Knowledge Graphs. AAAI 2016: 1955-1961

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, Guillaume Bouchard:
Complex Embeddings for Simple Link Prediction. ICML 2016: 2071-2080

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel:
Convolutional 2D Knowledge Graph Embeddings. AAAI 2018: 1811-1818

Timothée Lacroix, Nicolas Usunier, Guillaume Obozinski:
Canonical Tensor Decomposition for Knowledge Base Completion. ICML 2018: 2869-2878

Pasquale Minervini, Luca Costabello, Emir Muñoz, Vít Novácek, Pierre-Yves Vandenbussche:
Regularizing Knowledge Graph Embeddings via Equivalence and Inversion Axioms. ECML/PKDD (1) 2017: 668-683

Pasquale Minervini, Thomas Demeester, Tim Rocktäschel, Sebastian Riedel:
Adversarial Sets for Regularising Neural Link Predictors. UAI 2017

Maximilian Nickel, Xueyan Jiang, Volker Tresp:
Reducing the Rank in Relational Factorization Models by Including Observable Patterns. NIPS 2014: 1179-1187

Richard Evans, Edward Grefenstette:
Learning Explanatory Rules from Noisy Data. J. Artif. Intell. Res. 61: 1-64 (2018)

Tim Rocktäschel, Sebastian Riedel:
End-to-end Differentiable Proving. NeurIPS 2017: 3791-3803

Patrick Hohenecker, Thomas Lukasiewicz:
Ontology Reasoning with Deep Neural Networks. CoRR abs/1808.07980 (2018)

Bibliography

Pasquale Minervini, Matko Bosnjak, Tim Rocktäschel, Sebastian Riedel:
Towards Neural Theorem Proving at Scale. CoRR abs/1807.08204 (2018)

Leon Weber, Pasquale Minervini, Jannes Münchmeyer, Ulf Leser, Tim Rocktäschel:
NLProlog: Reasoning with Weak Unification for Question Answering in Natural Language. ACL (1)2019: 6151-6161

Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, Luc De Raedt:
DeepProbLog: Neural Probabilistic Logic Programming. NeurIPS 2018: 3753-3763

Luciano Serafini, Artur S. d'Avila Garcez:
Logic Tensor Networks: Deep Learning and Logical Reasoning from Data and Knowledge. CoRR abs/1606.04422 (2016)

Ivan Donadello, Luciano Serafini, Artur S. d'Avila Garcez:
Logic Tensor Networks for Semantic Image Interpretation. IJCAI 2017: 1596-1602

Andres Campero, Aldo Pareja, Tim Klinger, Josh Tenenbaum, Sebastian Riedel:
Logical Rule Induction and Theory Learning Using Neural Theorem Proving. CoRRabs/1809.02193

Georgina Peake, Jun Wang:
Explanation Mining: Post Hoc Interpretability of Latent Factor Models for Recommendation Systems. KDD 2018: 2060-2069

Arthur Colombini Gusmão, Alvaro Henrique Chaim Correia, Glauber De Bona, Fábio Gagliardi Cozman:
Interpreting Embedding Models of Knowledge Bases: A Pedagogical Approach. CoRR abs/1806.09504 (2018)

Iván Sánchez Carmona, Sebastian Riedel:
Extracting Interpretable Models from Matrix Factorization Models. CoCo@NIPS 2015

Vicente Iván Sánchez Carmona, Tim Rocktäschel, Sebastian Riedel, Sameer Singh:
Towards Extracting Faithful and Descriptive Representations of Latent Variable Models. AAAI Spring Symposia 2015

Thanks!

p.minervini@ucl.ac.uk