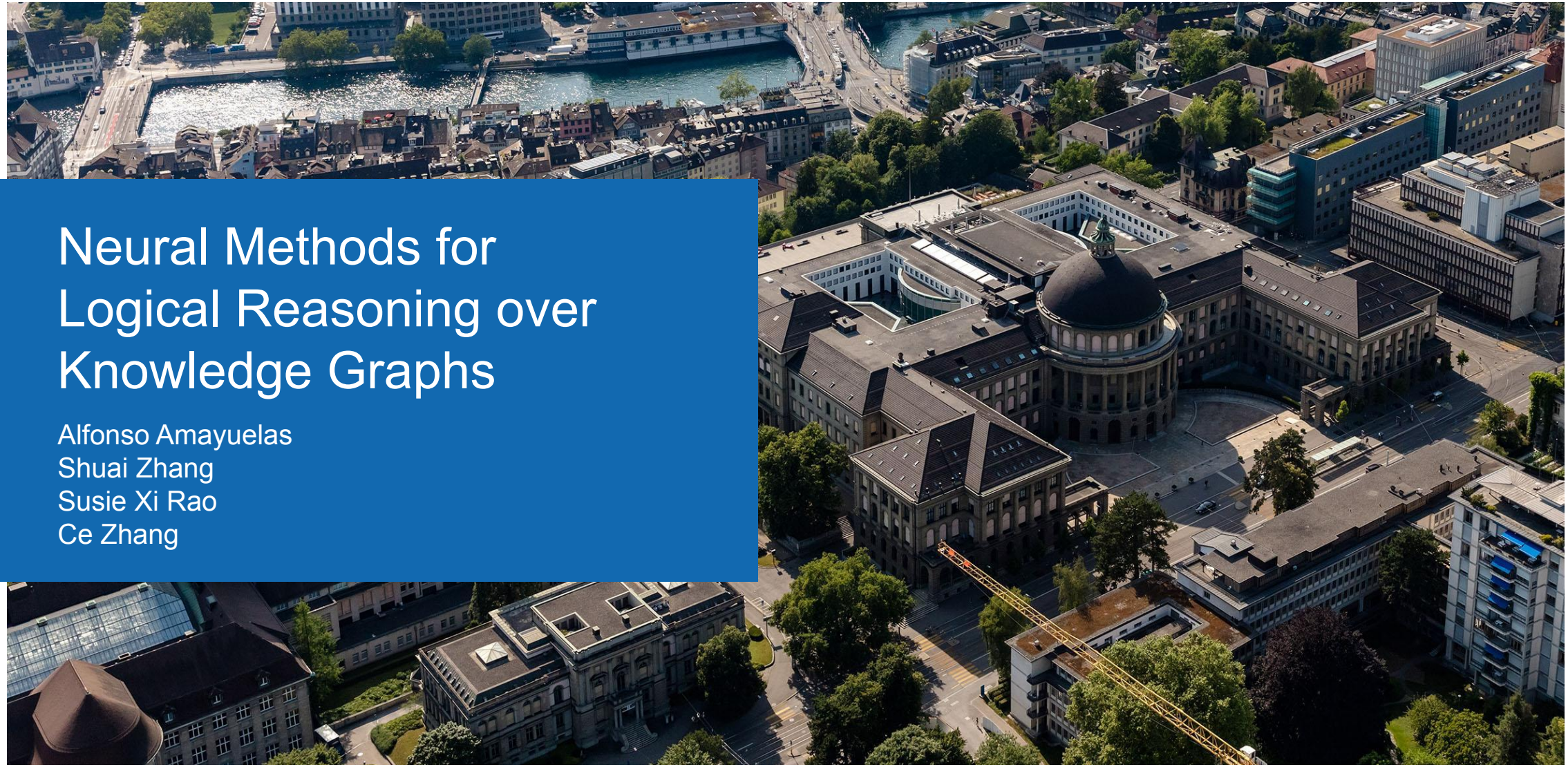
ETH zürich

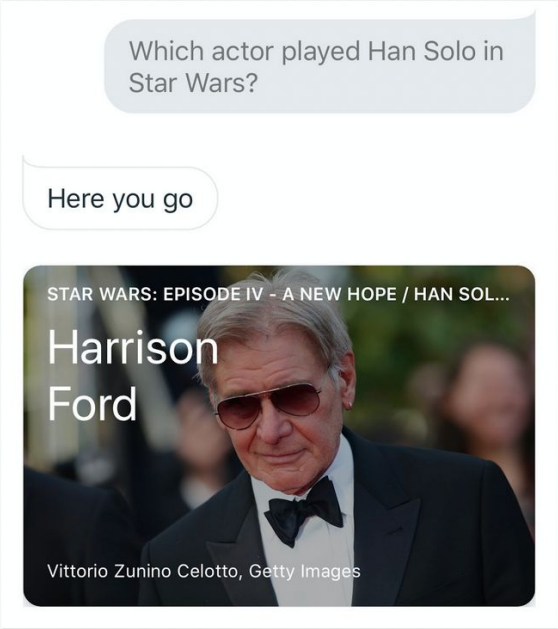
DS3Lab: Data Science, Data Systems, Data Services



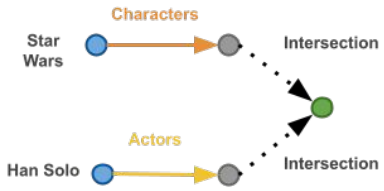
Neural Methods for Logical Reasoning over Knowledge Graphs

Alfonso Amayuelas
Shuai Zhang
Susie Xi Rao
Ce Zhang

But users want to ask more complex queries

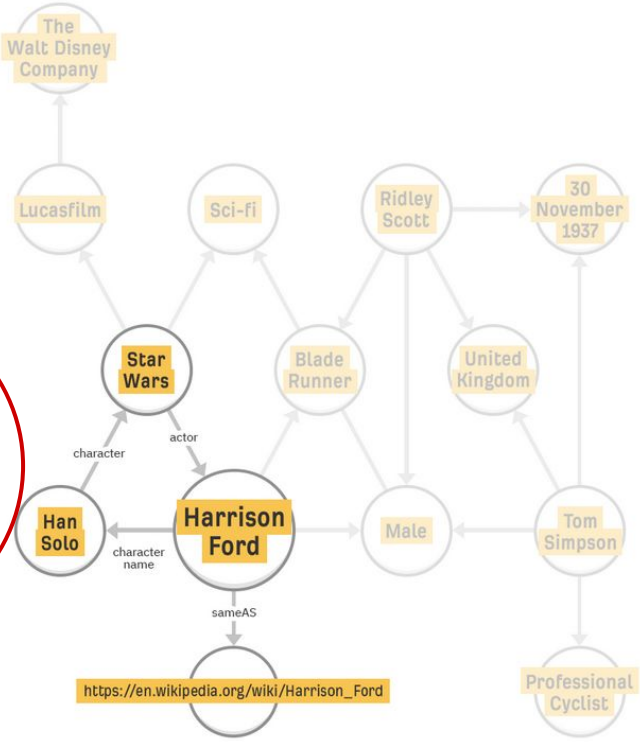


Natural Language Processing



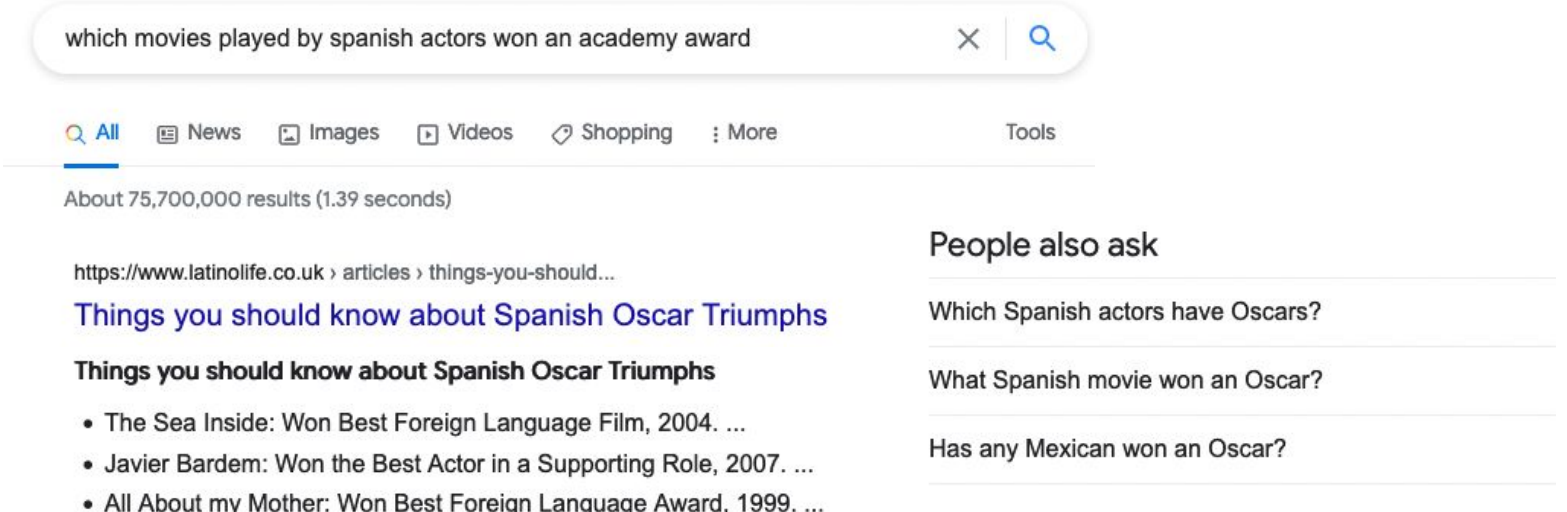
((Star Wars, characters) ∧ (Han Solo, playedBy))

Ask the Knowledge Graph



We will focus on this

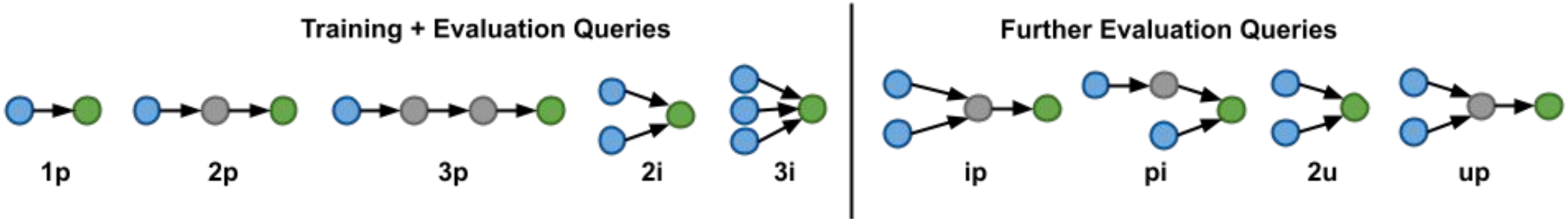
Here is where the problem starts...



Motivation: Logical Reasoning over Knowledge Graphs

Can we do multi-hop reasoning i.e. answer complex queries on an incomplete massive KG?

Our goal: Apply full First-order logic: *Existential, Negation, Conjunction, Disjunction*



Why is it complicated?

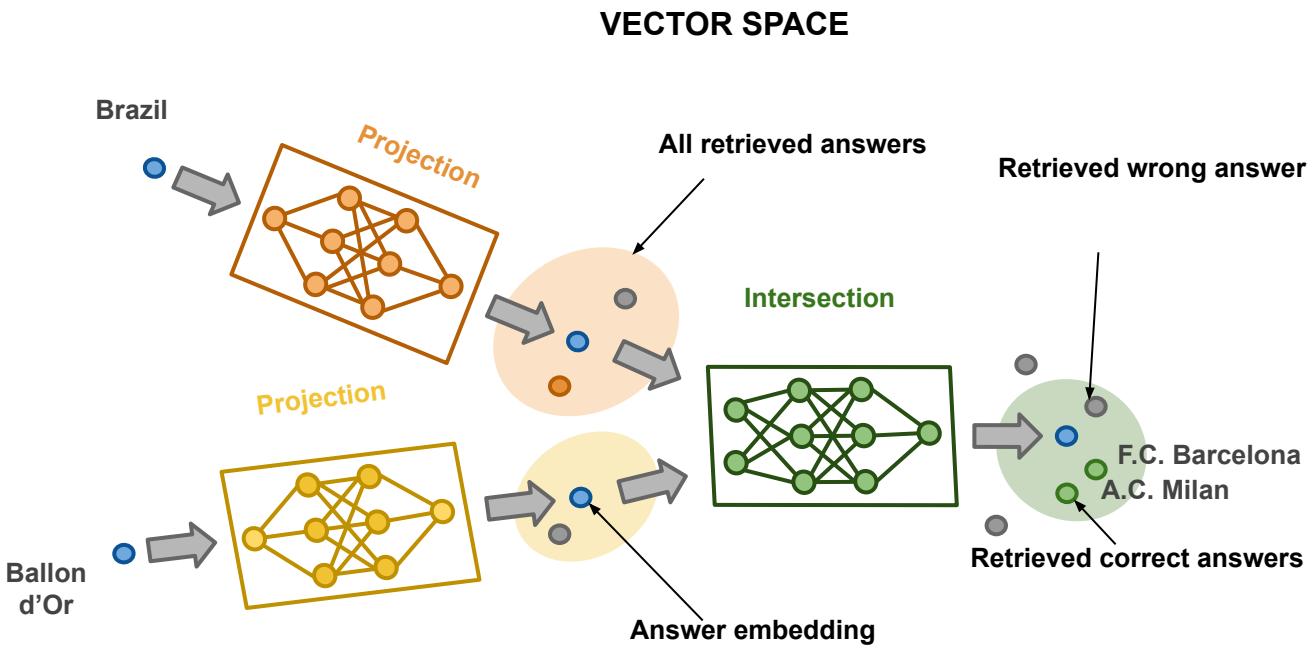
1. Increase in query-hops => Increase in complexity of traditional methods
2. Knowledge graphs are incomplete and very large in real cases

How the system works

Query: *"List the teams where Brazilian football players who were awarded a Ballon d'Or played"*

Input: Query in logical statements: *(((Brazil, CitizenOf), (Ballon d'Or, Winner)), Team)*

Output: Rank of all entities, according to the value of the negative random sampling. Done by near-neighbor search via *Locality Sensitivity Hashing*



The model uses low-dimensional one-point vector embeddings to represent the entities.

The logical operations are defined by Neural Networks

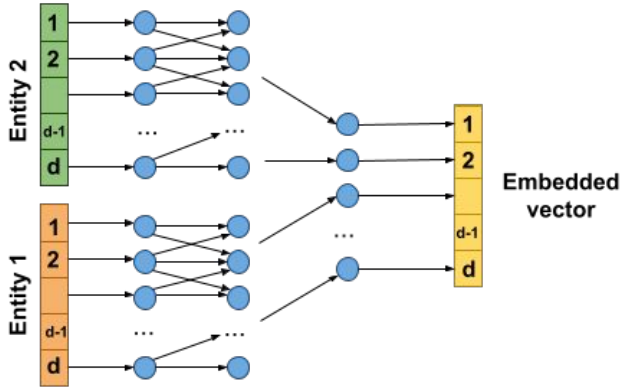
Multi-Layer Perceptron (**MLP**)

MLP: Operators are defined by a simple Multi-Layer Perceptron Neural Network

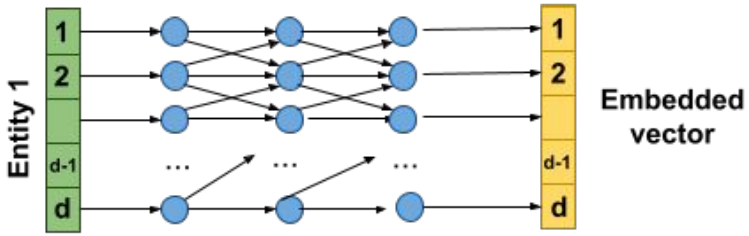
Why Neural Networks? Good results in basic KG tasks [5]

(+) Flexibility to “learn” any operator

(-) Hard to understand what they really learn



2-input operator



1-input operator

Operators:

- Projection $P(s_i, r_j) = NN_k(s_i, r_j), \forall i \in S, \forall j \in |R|$
- Intersection $I(s_i, s_j) = NN_k(s_i, s_j), \forall i, j \in S$
- Negation $N(s_i) = NN_k(s_i), \forall s_i \in S$

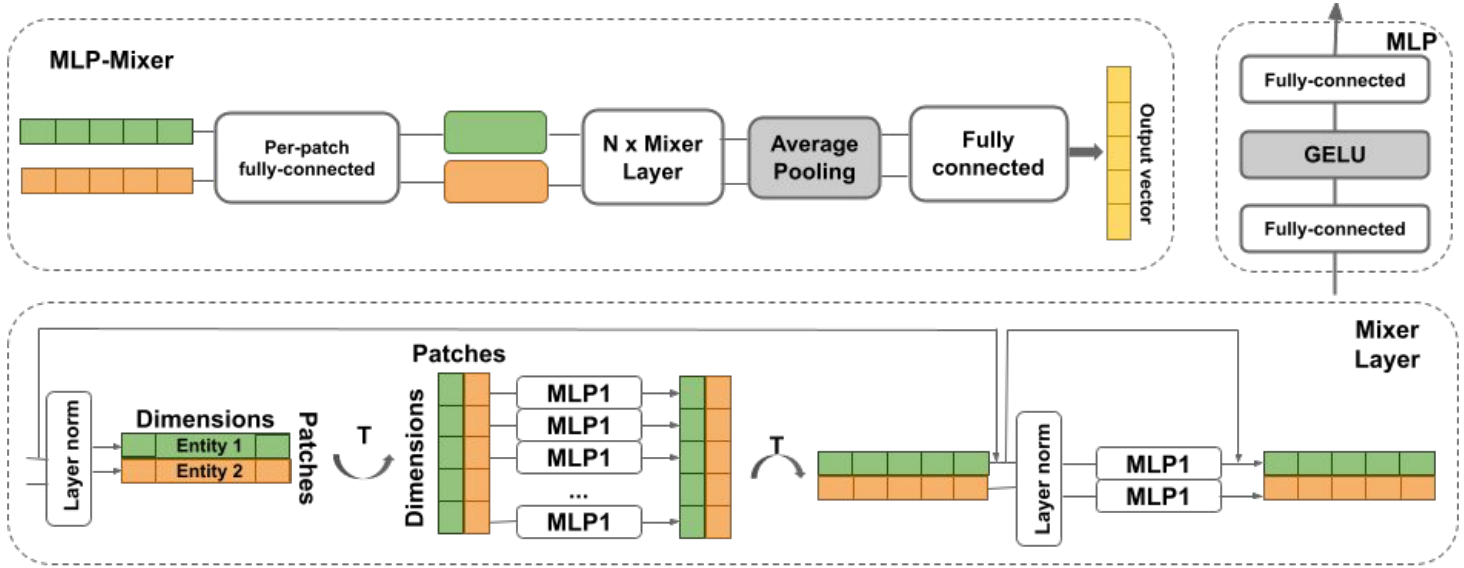
where s_i represents an entity variable and r_j a relation

!*Notice* we do not have a Union operator because we represent queries in DNF

Multi-Layer Perceptron Mixer (**MLP-Mixer**)

MLP-Mixer [6] is a Neural Network Architecture built for Image Processing with the purpose of removing convolutions.

MLP-Mixer Modules



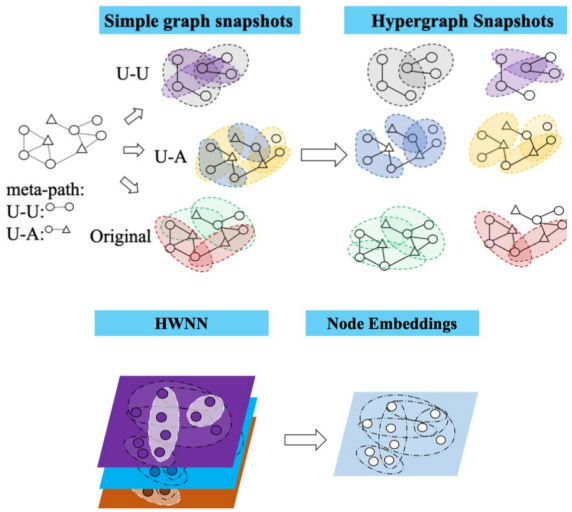
2-input Operators:

- Projection $P(s_i, r_j) = MLP_{mix}(s_i, r_j) \forall i \in S, \forall j \in |R|$
- Intersection $I(s_i, s_j) = MLP_{mix}(s_i, s_j), \forall i, j \in S$

Insight: Convolutional layers are not performing great -> Intuitively, nearby entities are not initially related

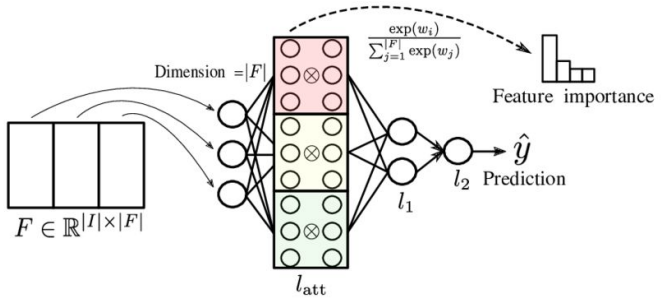
Model variants

Hyper-Graph Embeddings



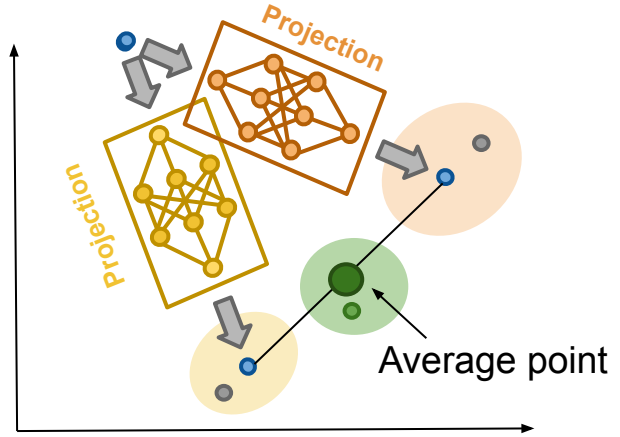
Graph embedding transformation into a series of snapshots to create new node embeddings

Attention mechanism



Include the attention layers in the operators to focus on the "important" parts of the embedding

2-vector Average



NNs may learn different "close" results

We compute the operators twice and average the result

Results: Models vs Baselines

Insights:

1. We see a considerable improvement in all cases
2. Neural Networks seem to be capturing the essence of the logical operations
3. A combination of model variants can potentially lead to a better improvement

	Best model so far (BetaE)			1-point vector encoding (GQE)		
Dataset	**MLP**	**MLP-Mixer**	**2-vector avg**	**MLP**	**MLP-Mixer**	**2-vector avg**
FB15k	5%	4%	9%	36%	35.5%	39%
FB15k-237	7%	9%	11%	28%	29%	31%
NELL995	7%	9%	8%	30%	32%	31%

Table of relative improvements (%)

Results: Negative Queries

Pros: Achieve better results than baseline model

Cons: (1) Hard to understand the representation of a negative entity
(2) Unable to measure query uncertainty

Dataset	Model	2in	3in	inp	pin	pni	avg
FB15K	MLP	**17.2**	**17.8**	**13.5**	**9.1**	**15.2**	**14.5**
	BetaE	14.3	14.7	11.5	6.5	12.4	11.8
FB15-237	MLP	**6.6**	**10.7**	**8.1**	**4.7**	**4.4**	**6.9**
	BetaE	5.1	7.9	7.4	3.6	3.4	5.4
NELL995	MLP	**5.1**	**8.0**	**10.0**	**3.6**	**3.6**	**6.1**
	BetaE	**5.1**	7.8	**10.0**	3.1	3.5	5.9

19% improvement

22% improvement

4% improvement

(Measure) **MRR: Mean Reciprocal Rank**

Future work

Some points to cover in future work:

- **Logical Disjunction (∨)**: Now we use DNF transformation to handle this query. Look into other methods that create an OR operator.
- **Logic Regularizers**: Logic Neural Networks can be a good fit for this problem. Our initial results seemed discouraging. Further research needed.
- **Query/answer generation**: The structures we use for training and testing may have a great influence on the system performance. Better query generation techniques could have a performance boost.
- **Other embeddings**: There are other types of embeddings unexplored for this problem. Look into other promising graph embeddings (i.e. quantum embeddings)

References

[1] J.Leskovec, CS224W: Machine Learning with Graphs. Stanford. Winter 2021. Available: http://web.stanford.edu/class/cs224w/

[2] Hamilton, W. L., Bajaj, P., Zitnik, M., Jurafsky, D., & Leskovec, J. (2018). Embedding logical queries on knowledge graphs. arXiv preprint arXiv:1806.01445.

[3] Ren, Hongyu, Weihua Hu, and Jure Leskovec. "Query2box: Reasoning over knowledge graphs in vector space using box embeddings." arXiv preprint arXiv:2002.05969 (2020)

[4] Ren, Hongyu, and Jure Leskovec. "Beta embeddings for multi-hop logical reasoning in knowledge graphs." arXiv preprint arXiv:2010.11465 (2020).

[5] Socher, Richard, et al. "Reasoning with neural tensor networks for knowledge base completion." Advances in neural information processing systems. 2013.

[6] Tolstikhin, Ilya, et al. "Mlp-mixer: An all-mlp architecture for vision." arXiv preprint arXiv:2105.01601 (2021).

Thank you

Alfonso Amayuelas Fernández
alfonso.amayuelas.alumni@epfl.ch

ETH zürich

DS3Lab: Data Science, Data Systems, Data Services

Supervisors:
Shuai Zhang
Susie Rao

Lab Director:
Ce Zhang

https://ds3lab.inf.ethz.ch

EPFL

Distributed Information Systems Laboratory

Lab Director:
Karl Aberer

https://www.epfl.ch/labs/lsir/