Injecting Logical Background Knowledge into Embeddings for Relation Extraction

Tim Rocktäschel, Sameer Singh and Sebastian Riedel

t.rocktaschel@cs.ucl.ac.uk

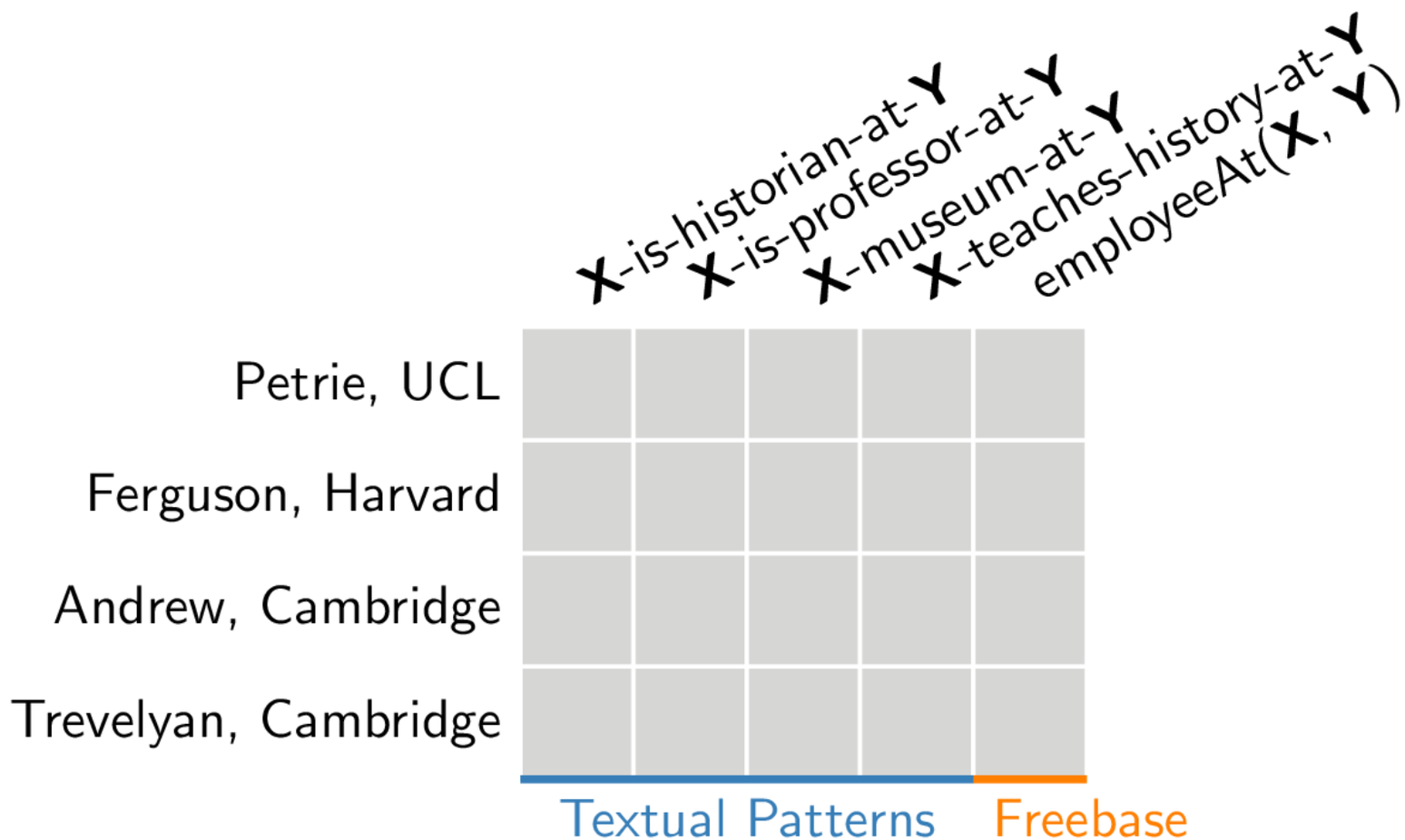
2nd of June 2015
NAACL, Denver

Freebase is incomplete

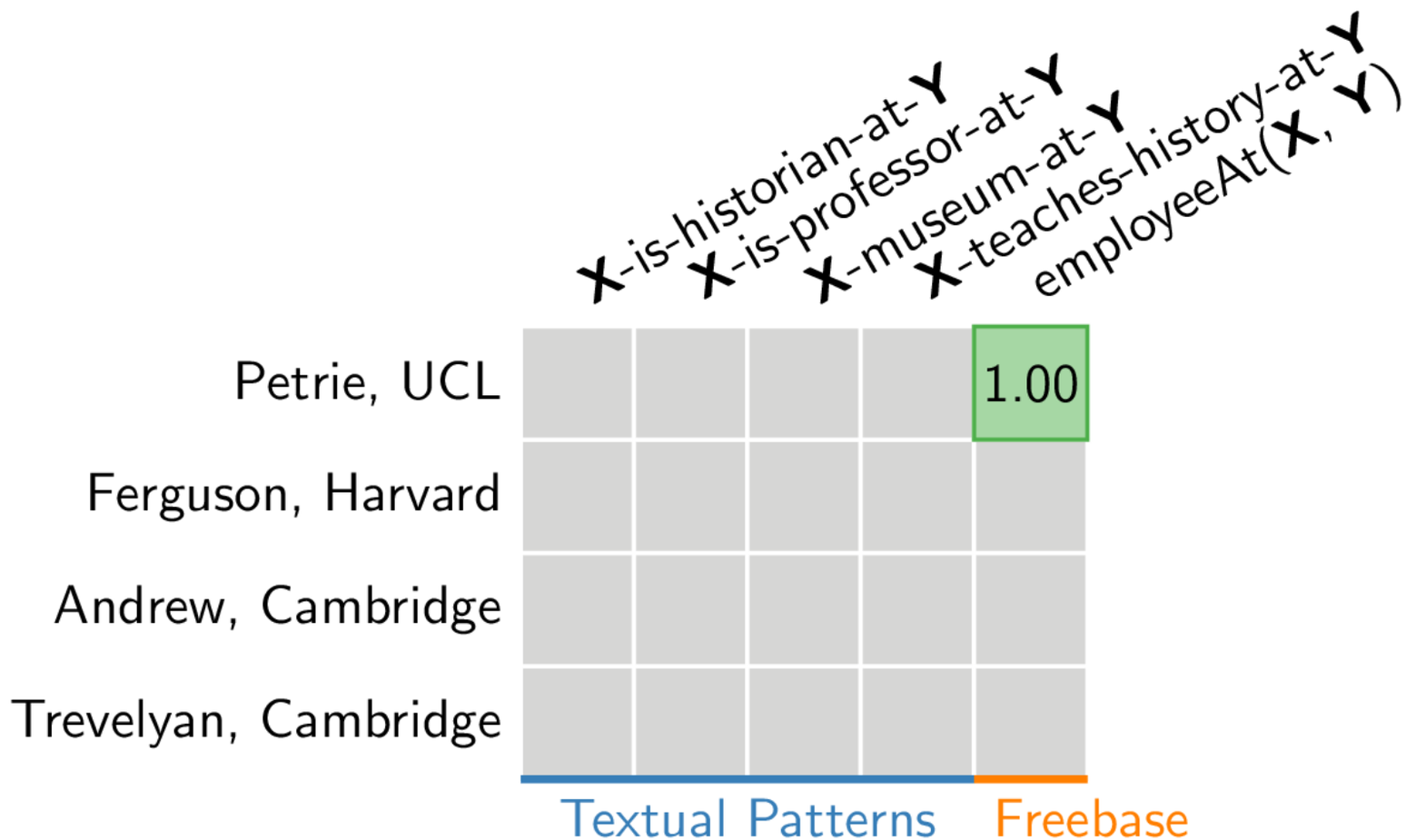
- **Missing Facts**: `placeOfBirth` attribute is missing for 71% of the people (Dong et al., 2014)
- **Missing Entities**: Contains no information about `UCL Machine Reading Lab`
- **Missing Relations**: May contain `profAt(John Shawe-Taylor, UCL)` but not `givesLecturesAt(John Shawe-Taylor, UCL)`
- **Machine reading and reasoning to the rescue!**

Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)

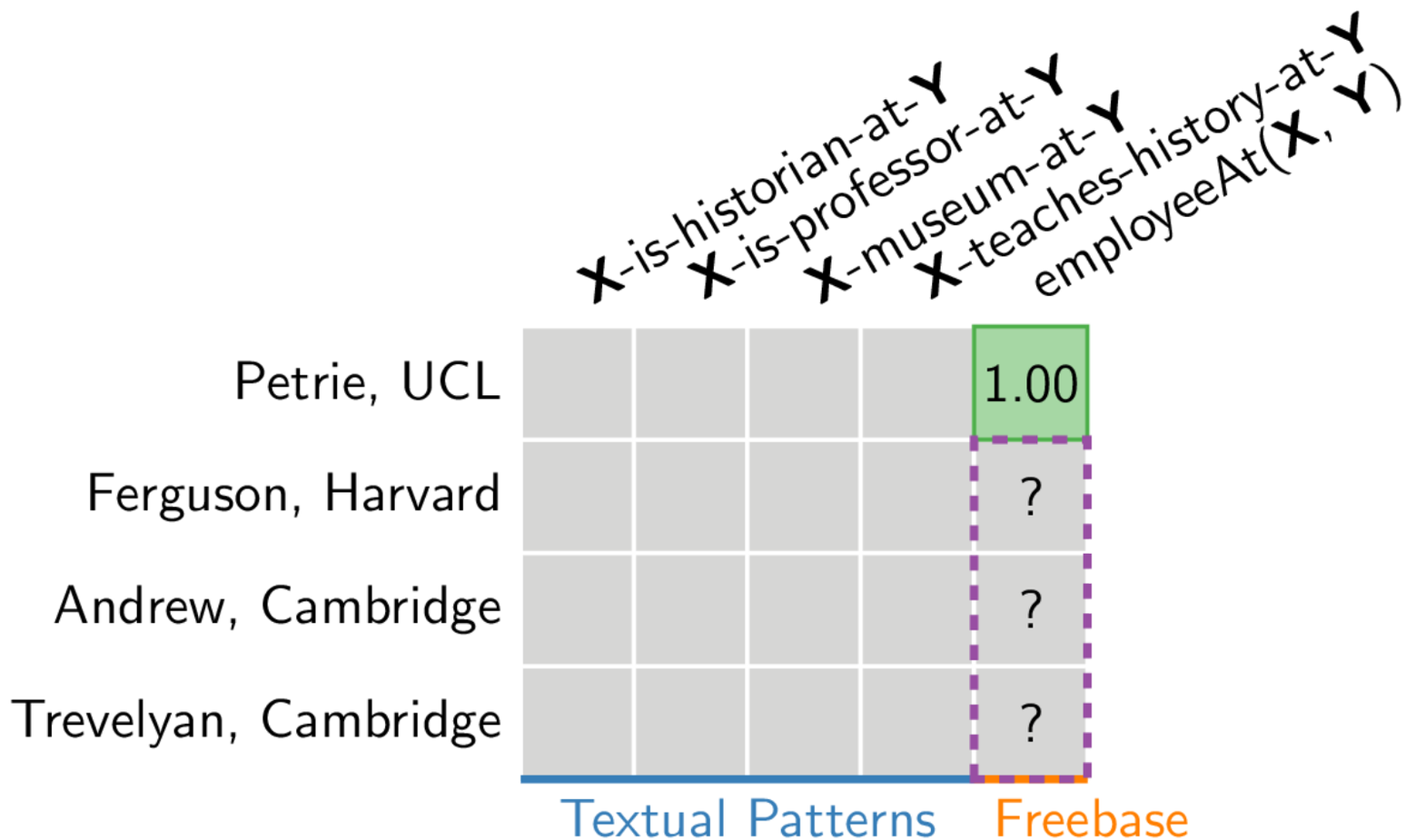
Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)



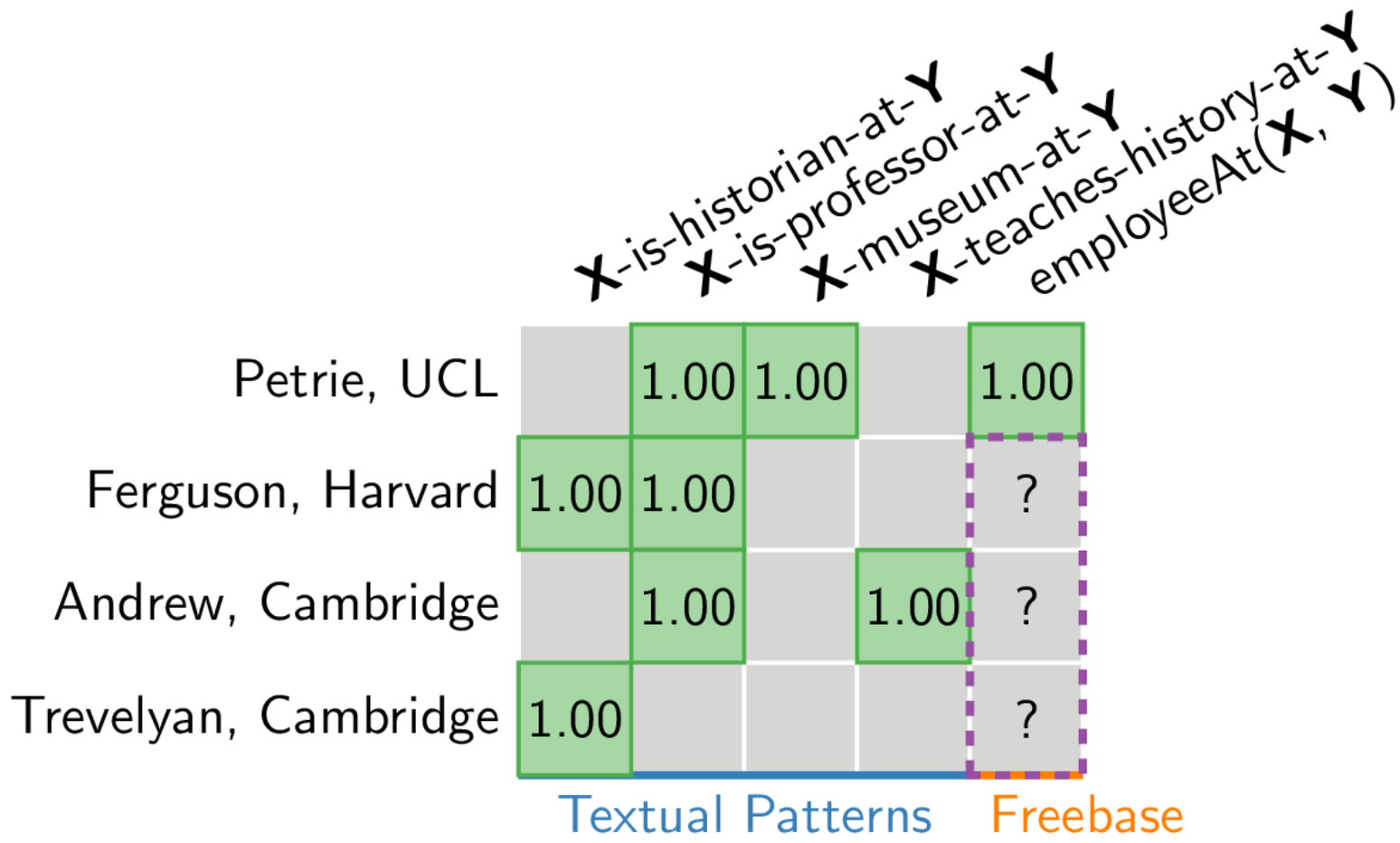
Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)



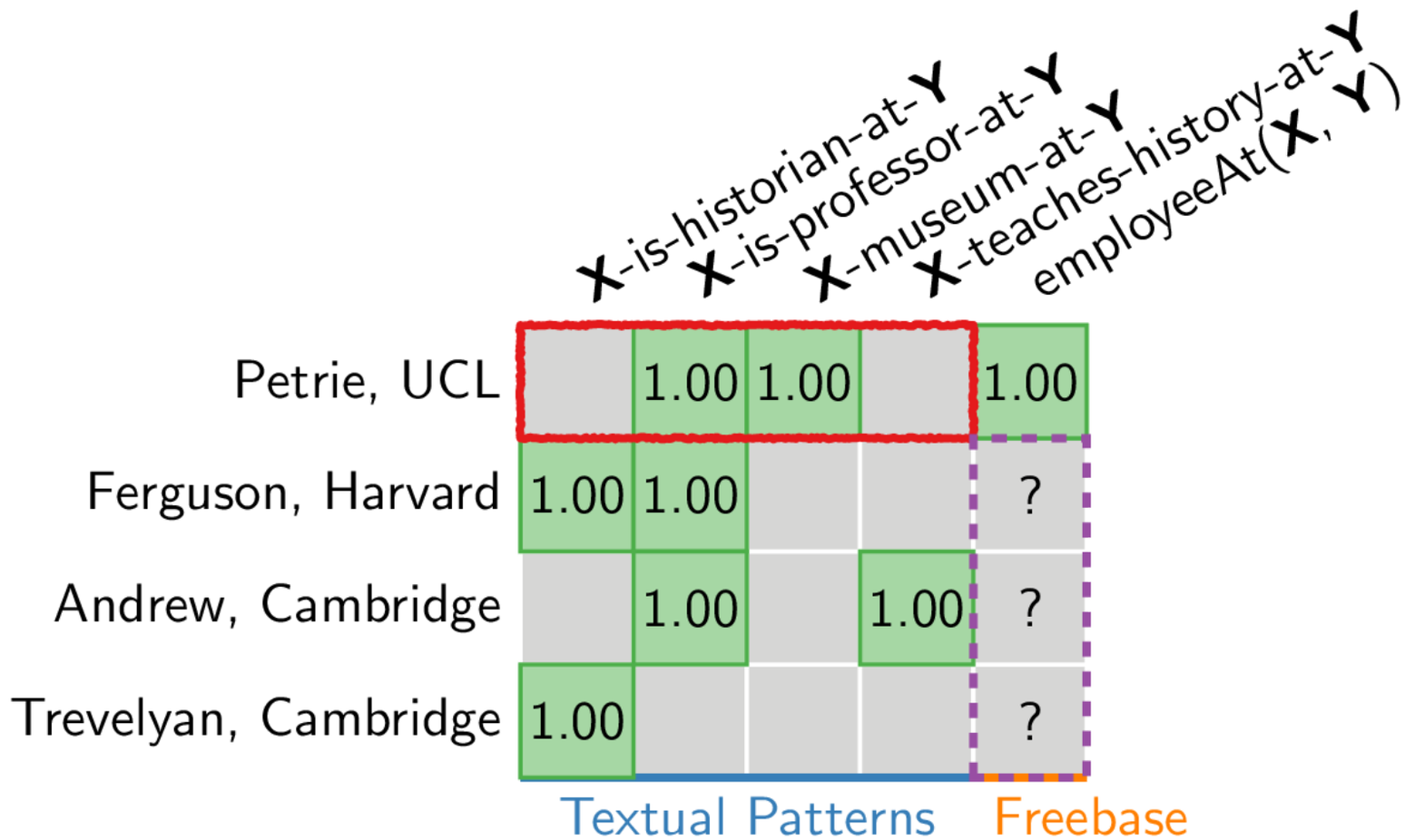
Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)



Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)



Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)

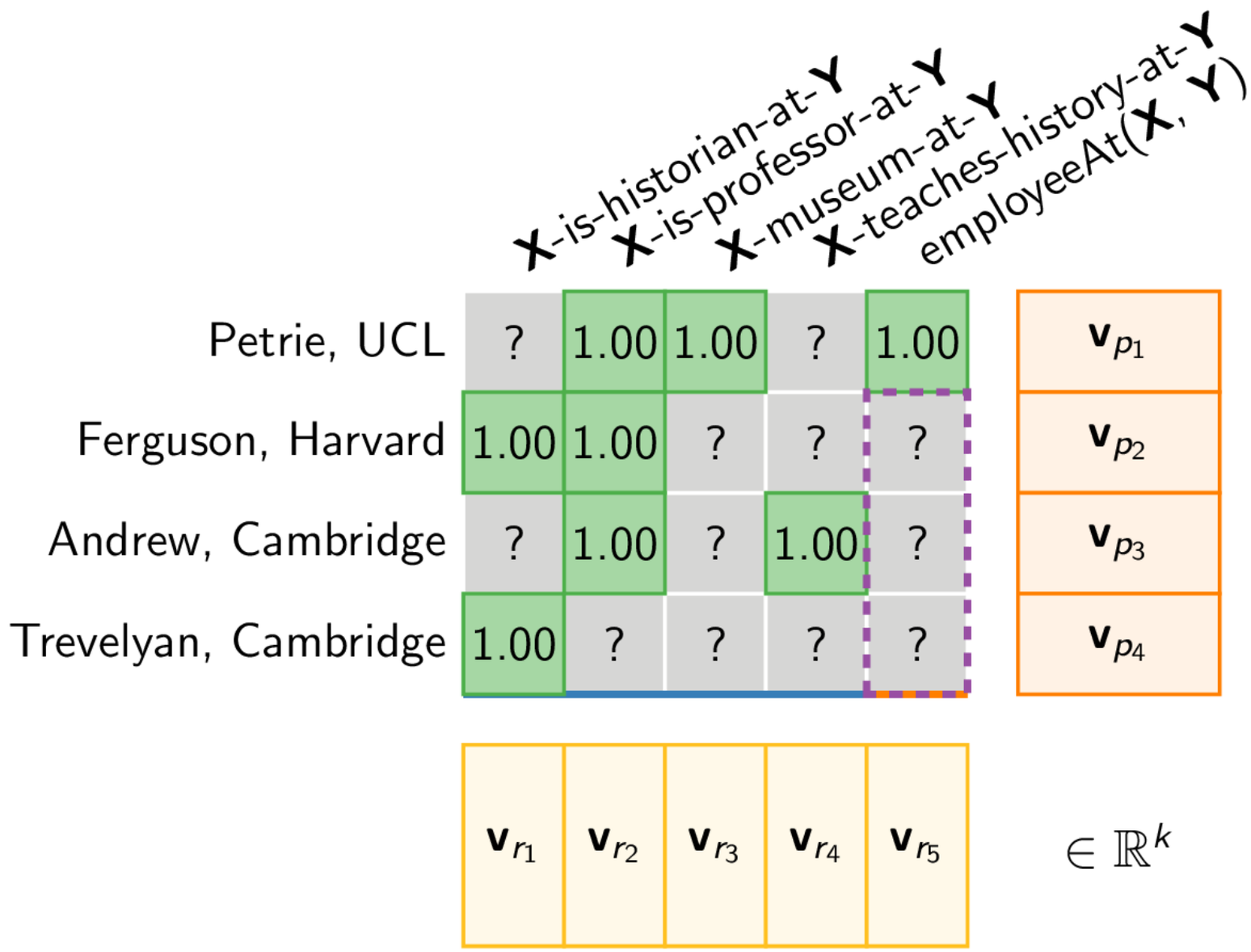


Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)

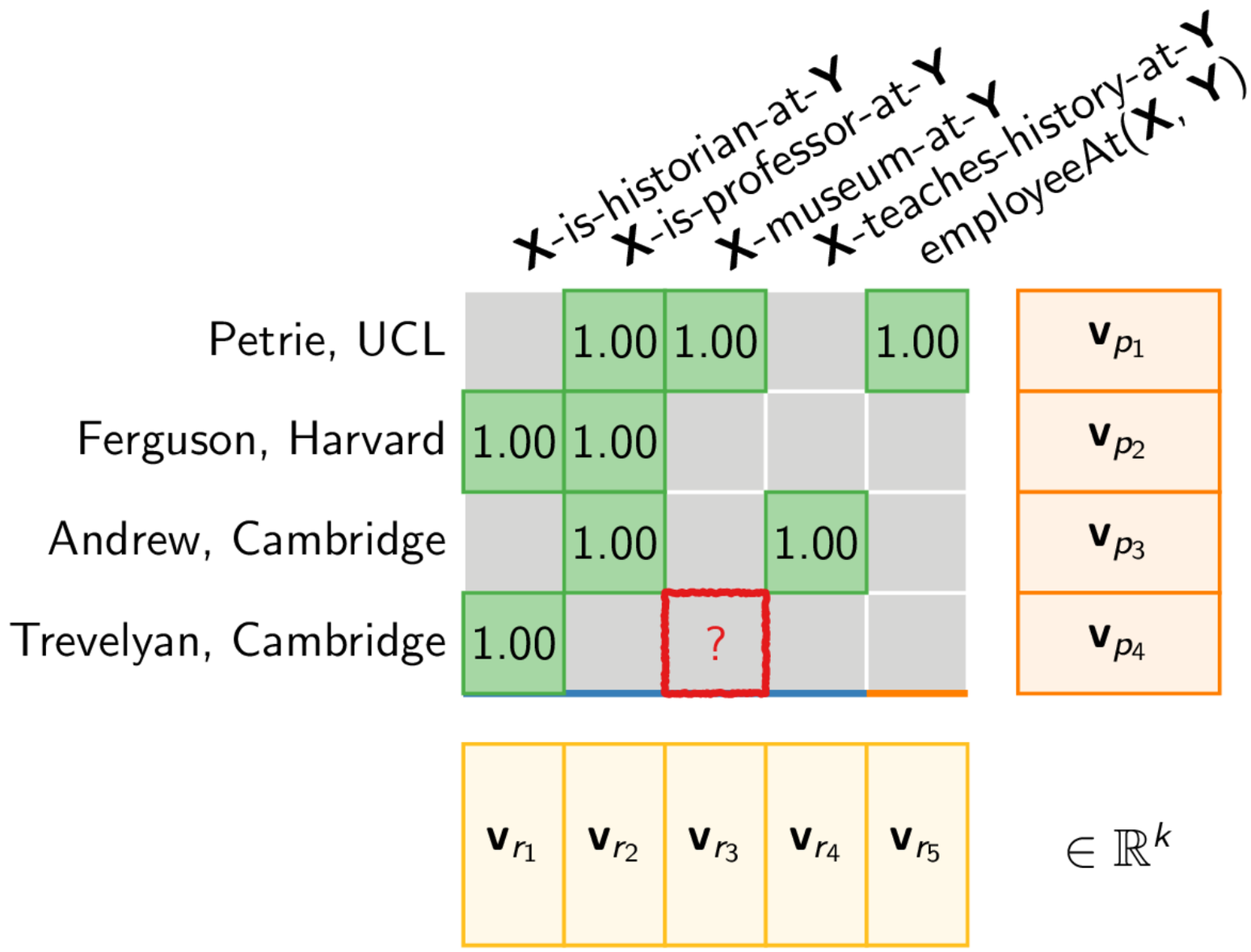
	X-is-historian-at-Y	X-is-professor-at-Y	X-museum-at-Y	X-teaches-history-at-Y	employeeAt(X, Y)
Petrie, UCL	?	1.00	1.00	?	1.00
Ferguson, Harvard	1.00	1.00	?	?	?
Andrew, Cambridge	?	1.00	?	1.00	?
Trevelyan, Cambridge	1.00	?	?	?	?

Textual Patterns Freebase

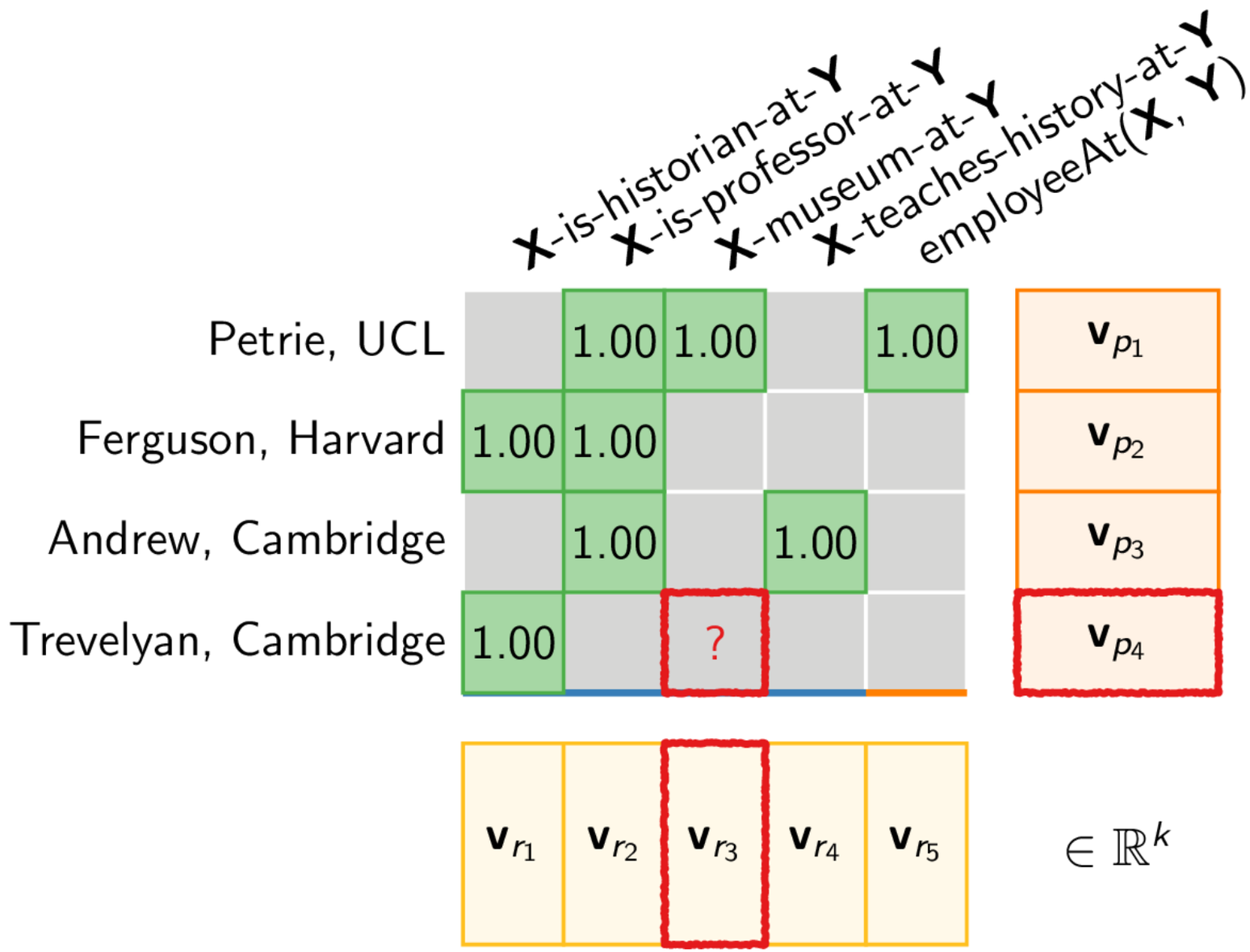
Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)



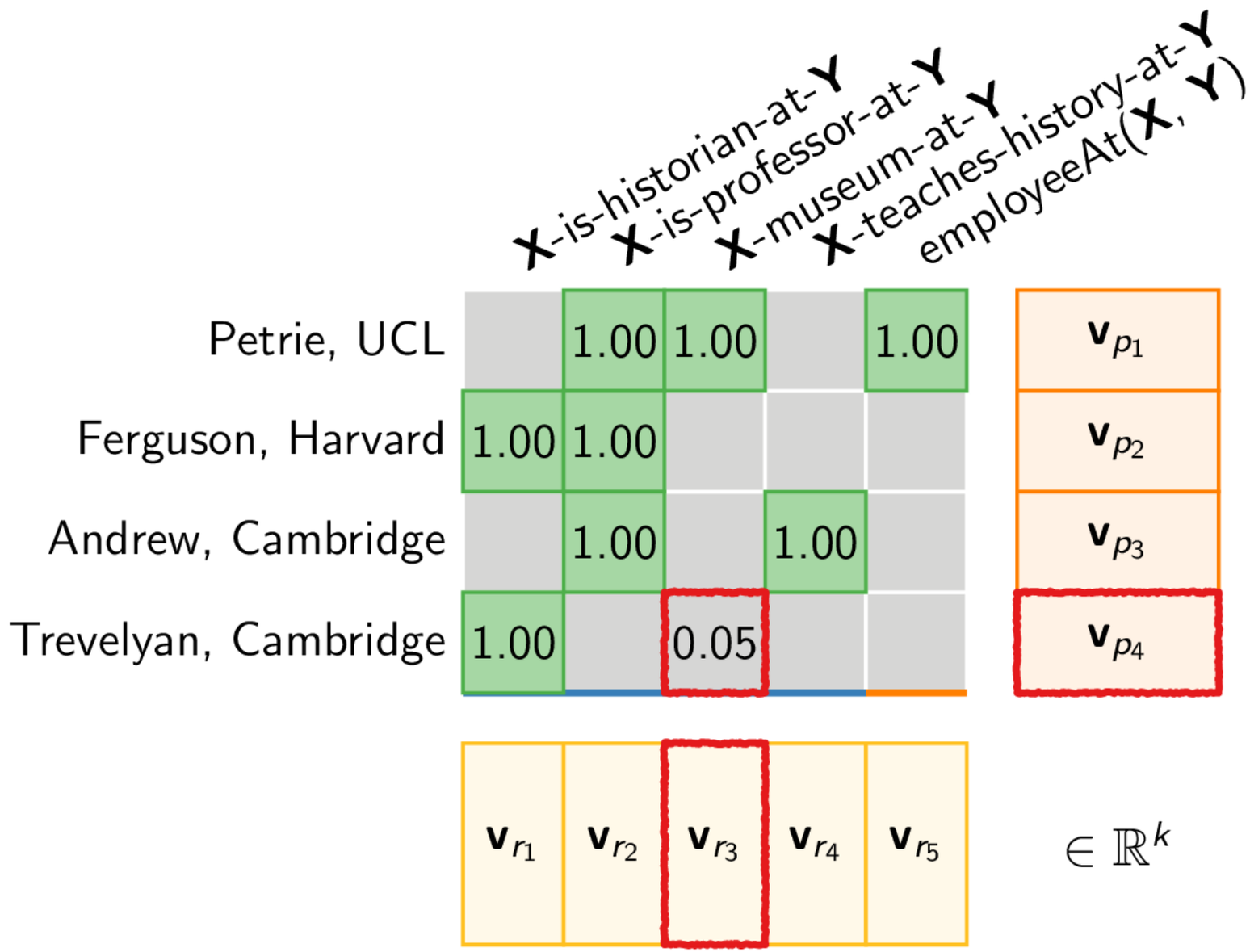
Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)



Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)



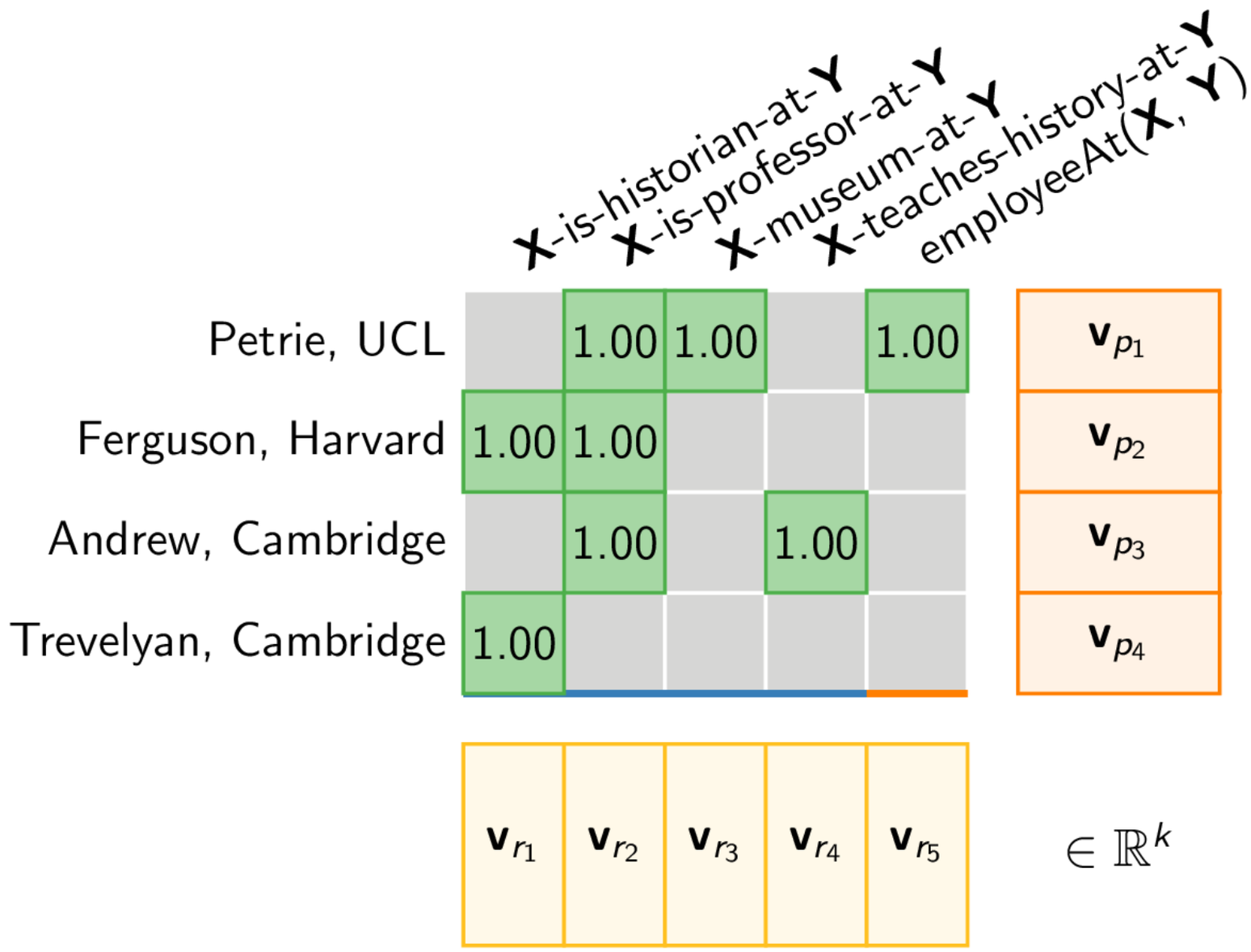
Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)



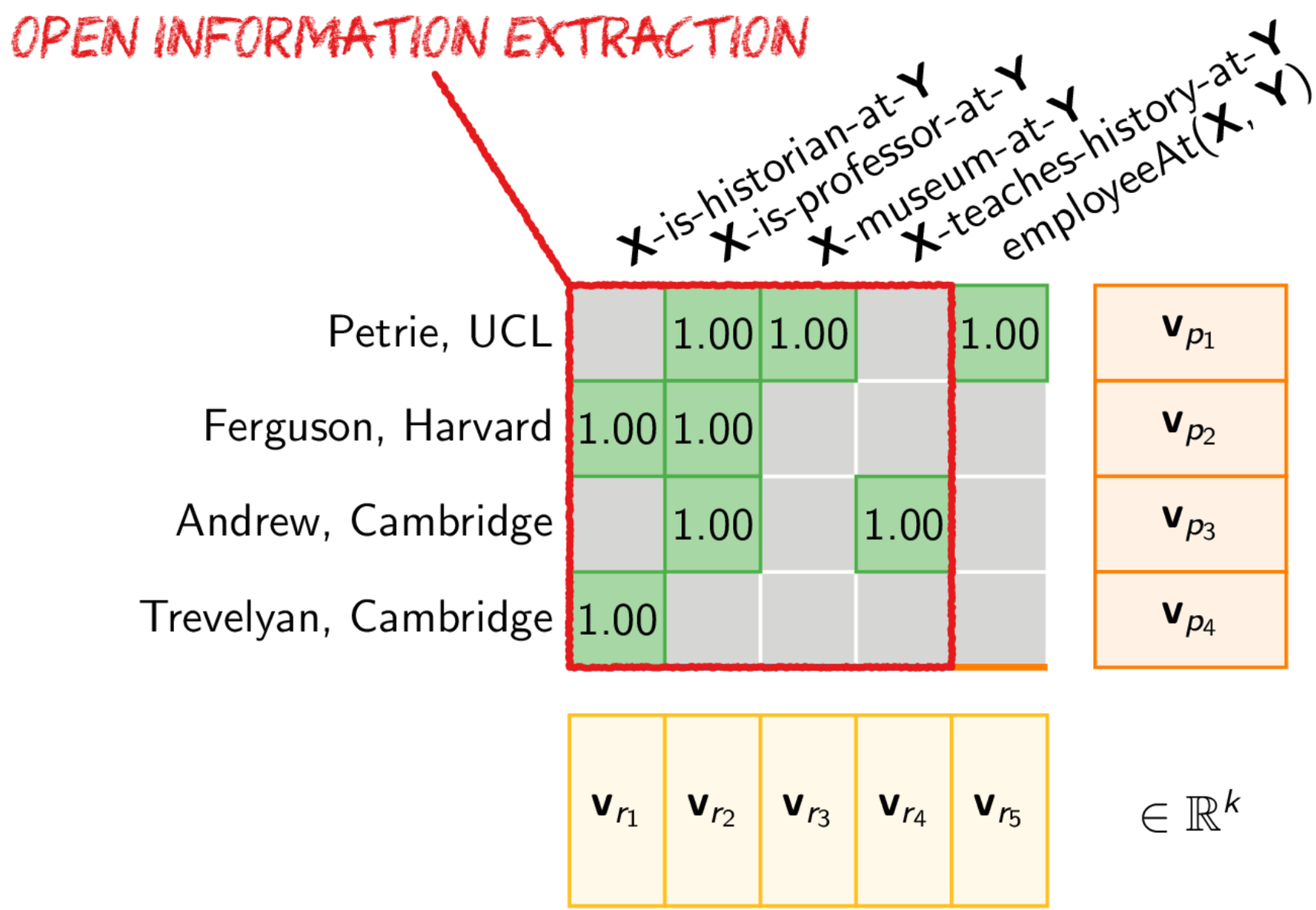
Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)

	X-is-historian-at-Y	X-is-professor-at-Y	X-museum-at-Y	X-teaches-history-at-Y	employeeAt(X, Y)	
Petrie, UCL	0.06	0.97	0.93	0.07	0.96	$\mathbf{v}_{p_1}$
Ferguson, Harvard	0.93	0.94	0.03	0.06	0.88	$\mathbf{v}_{p_2}$
Andrew, Cambridge	0.00	0.95	0.10	0.95	0.76	$\mathbf{v}_{p_3}$
Trevelyan, Cambridge	0.96	0.03	0.00	0.06	0.96	$\mathbf{v}_{p_4}$
	$\mathbf{v}_{r_1}$	$\mathbf{v}_{r_2}$	$\mathbf{v}_{r_3}$	$\mathbf{v}_{r_4}$	$\mathbf{v}_{r_5}$	$\in \mathbb{R}^k$

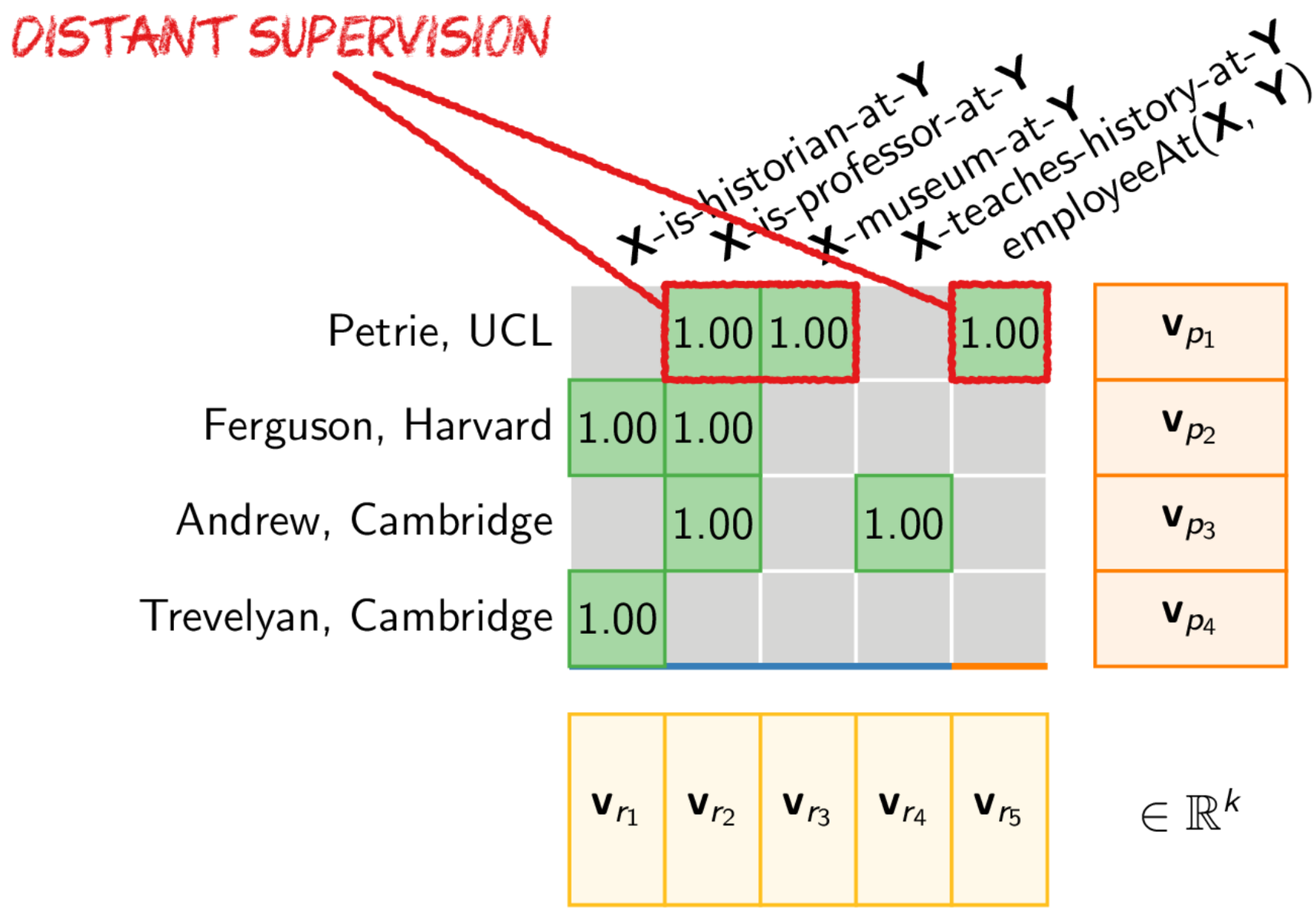
Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)



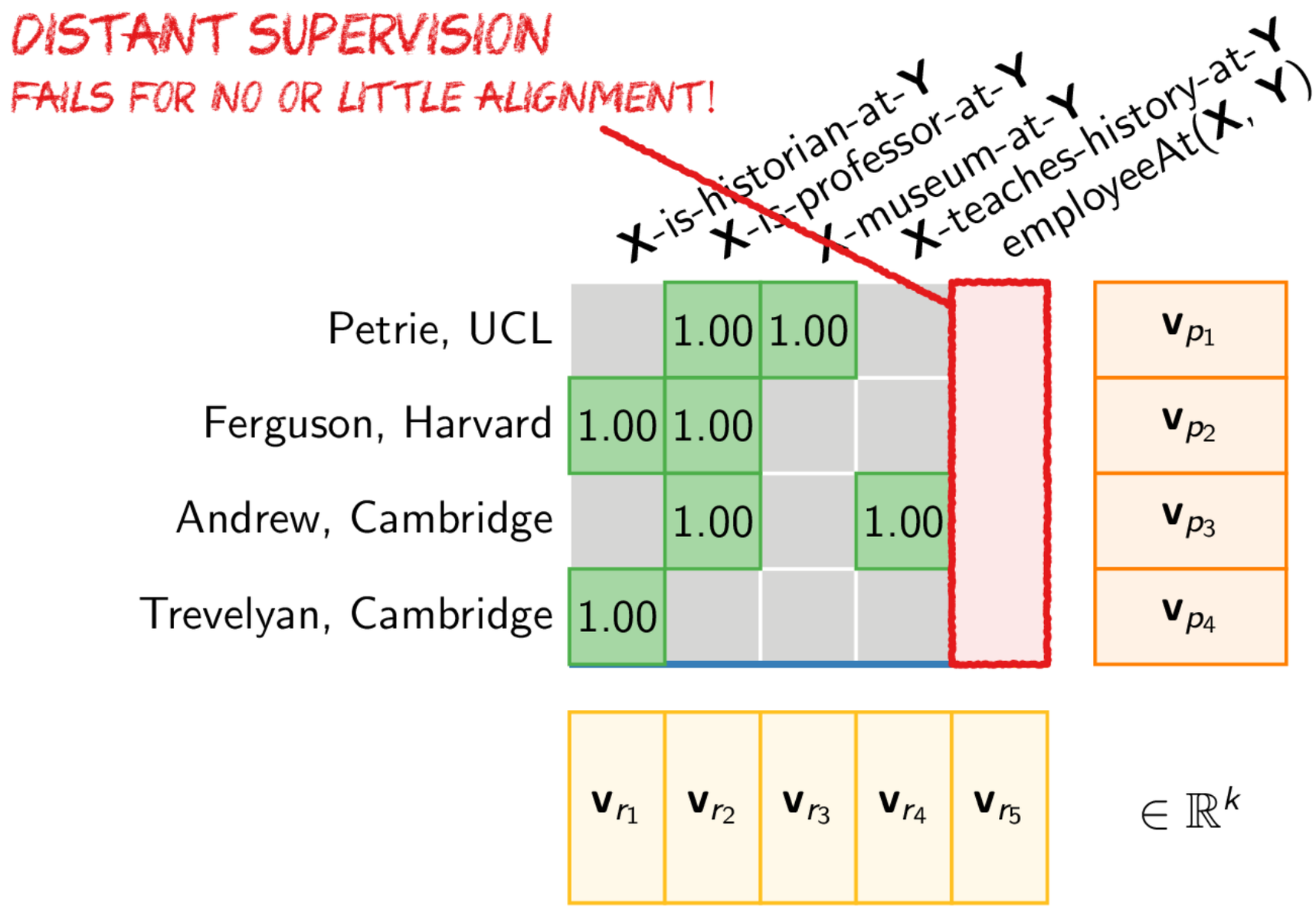
Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)



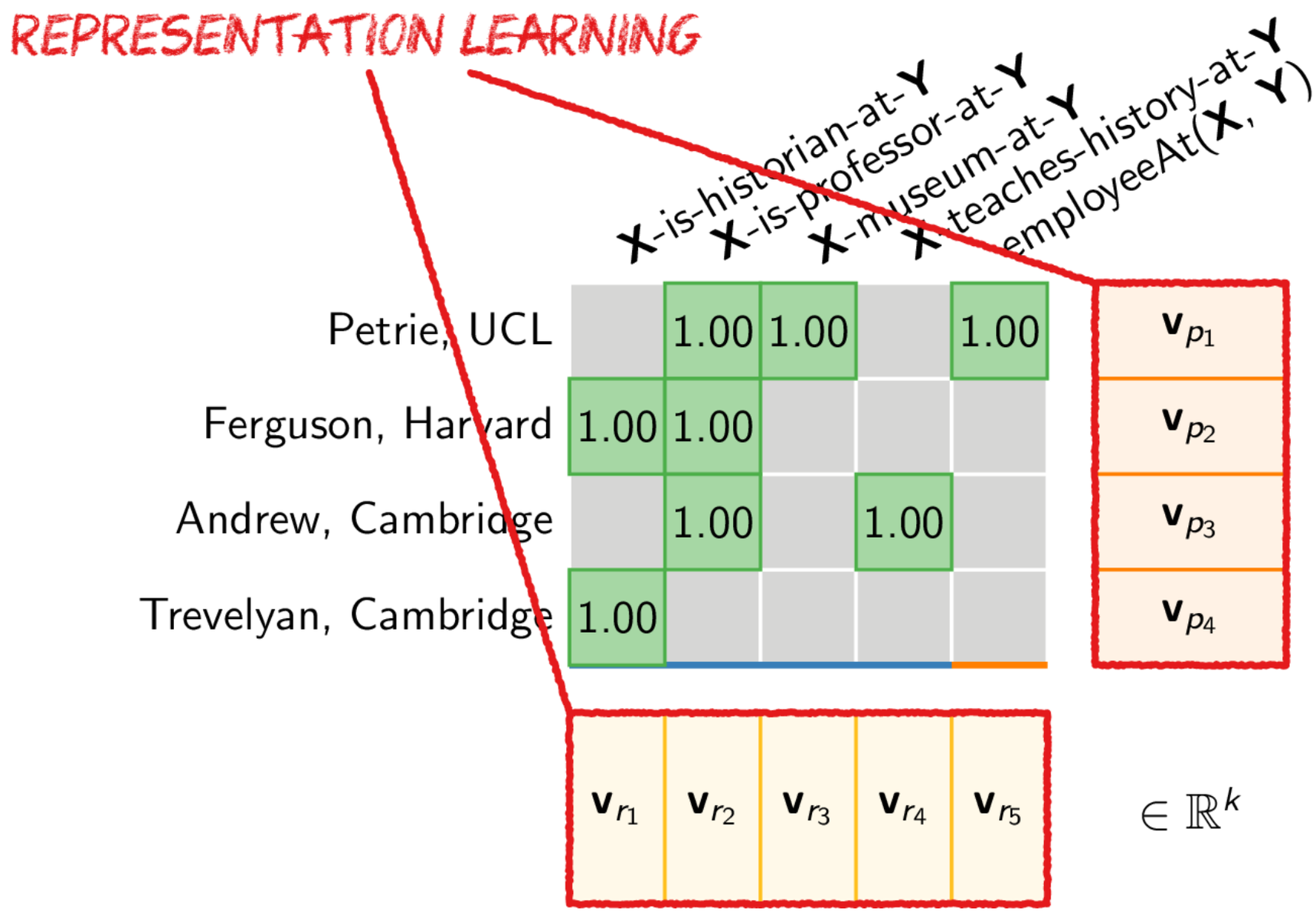
Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)



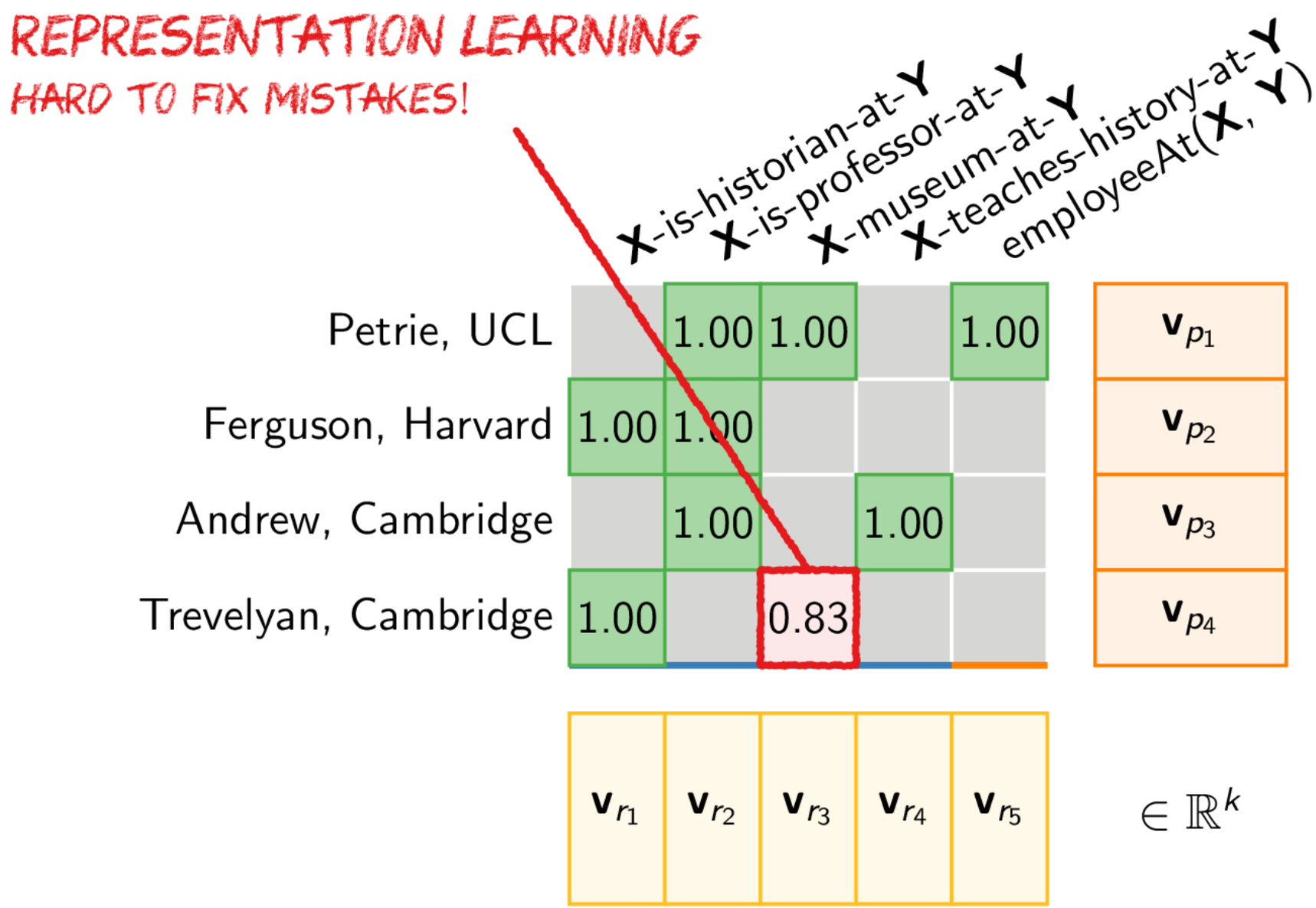
Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)



Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)



Relation Extraction with Matrix Factorization and Universal Schemas (Riedel et al., 2013)



We can fix this with logical formulae!

- **Representation Learning**: hard to fix mistakes

  $\neg$`person(x)` $\Rightarrow \neg$`place of birth(x,y)`

- **Distant Supervision**: fails for no or little alignment

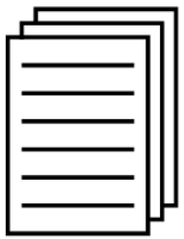
  `#1-is-a-student-in-#2-'s-lab(x,y)` $\Rightarrow$ `supervisedBy(x,y)`

- Pros: Formulae are easy to modify and improve
- Cons: Brittle, no generalization and inference can become intractable
- Markov Logic Networks: easy to modify, generalize well, but inference often intractable in practice

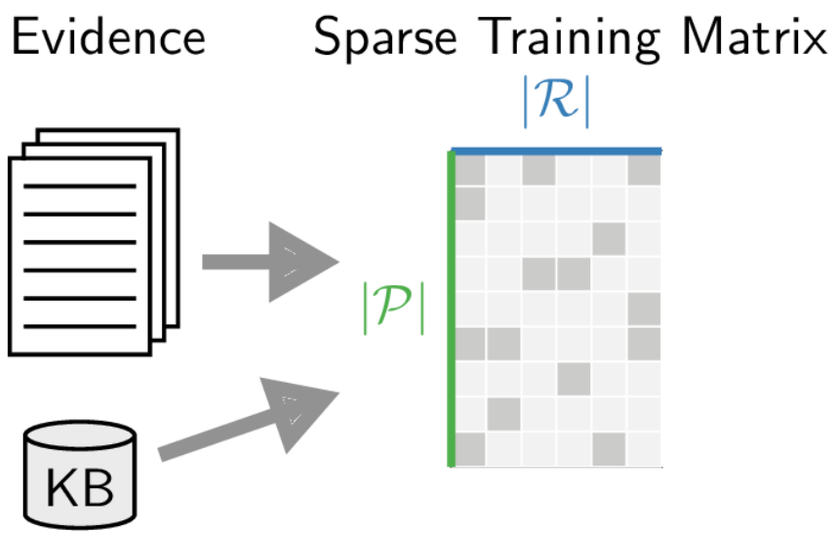
Overview

Overview

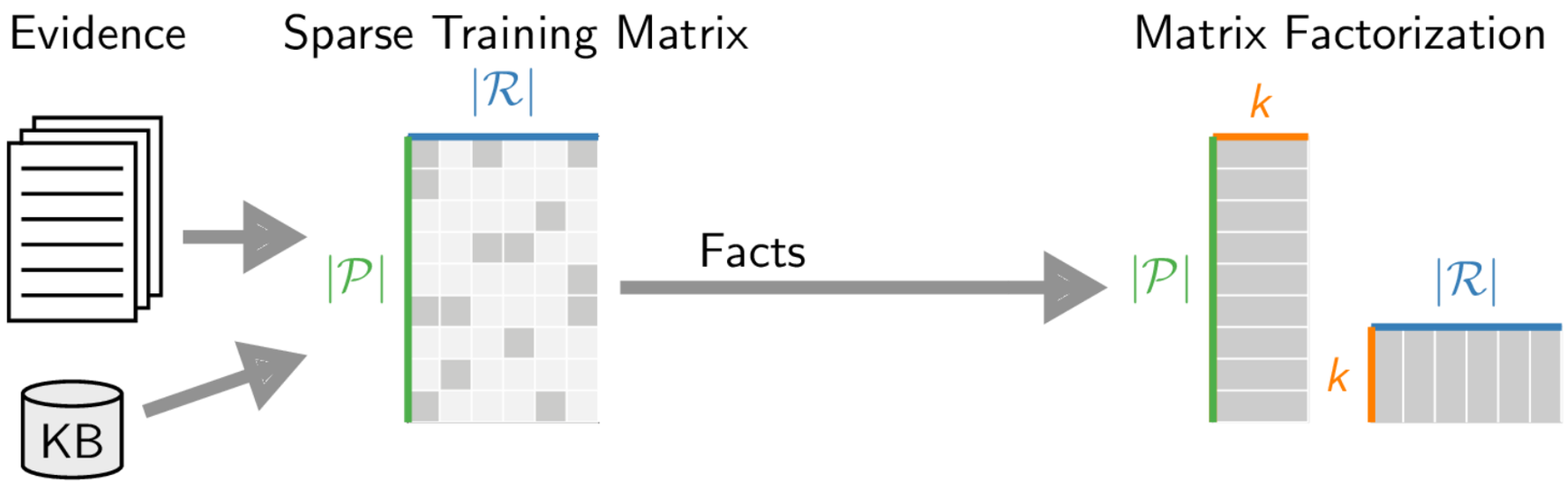
Evidence



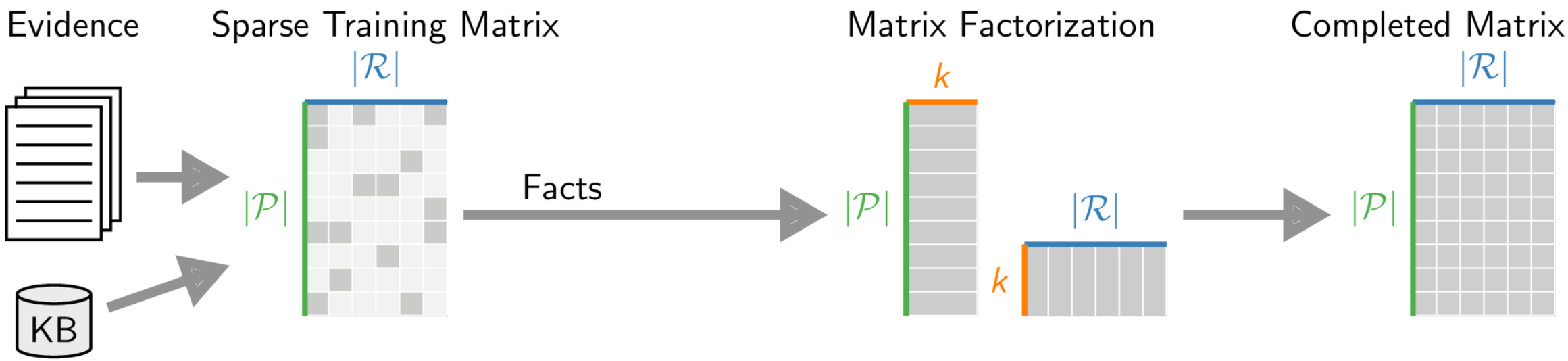
Overview



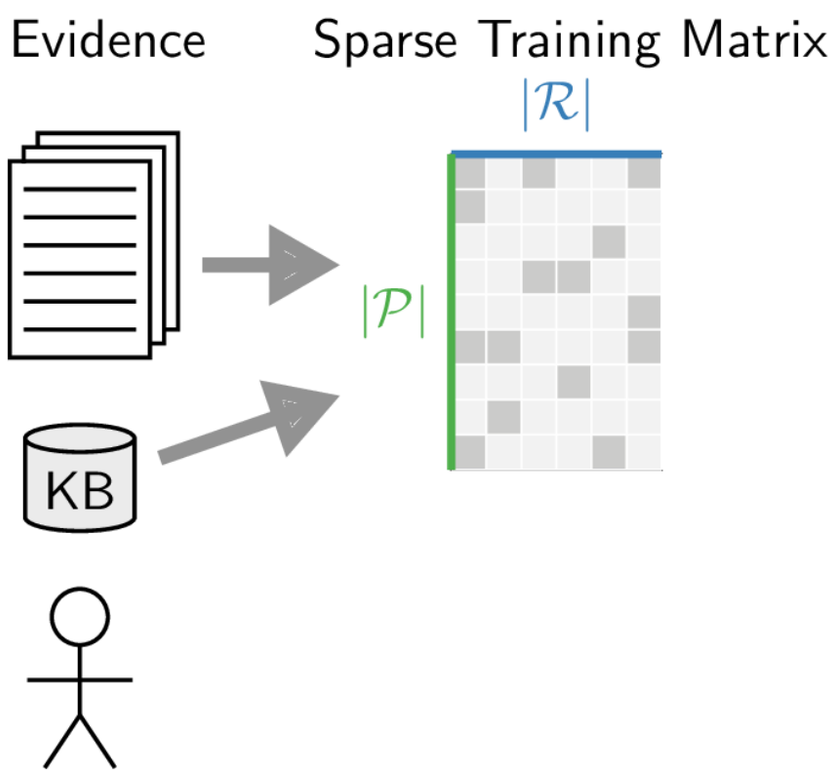
Overview


Evidence
Sparse Training Matrix
Matrix Factorization
$|\mathcal{R}|$
$|\mathcal{P}|$
Facts
k
$|\mathcal{P}|$
$|\mathcal{R}|$
k
KB

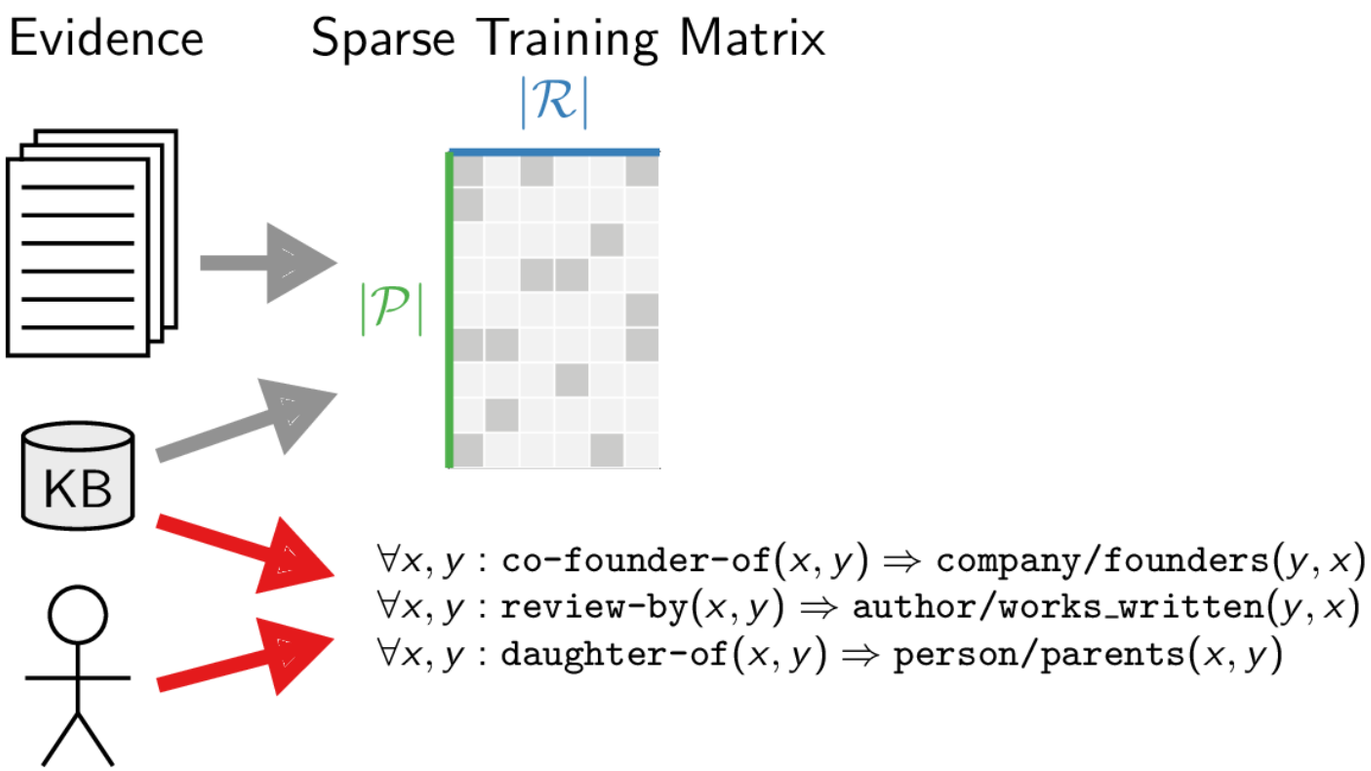
Overview


Evidence
Sparse Training Matrix
Matrix Factorization
Completed Matrix
$|\mathcal{R}|$
$|\mathcal{P}|$
Facts
k
KB

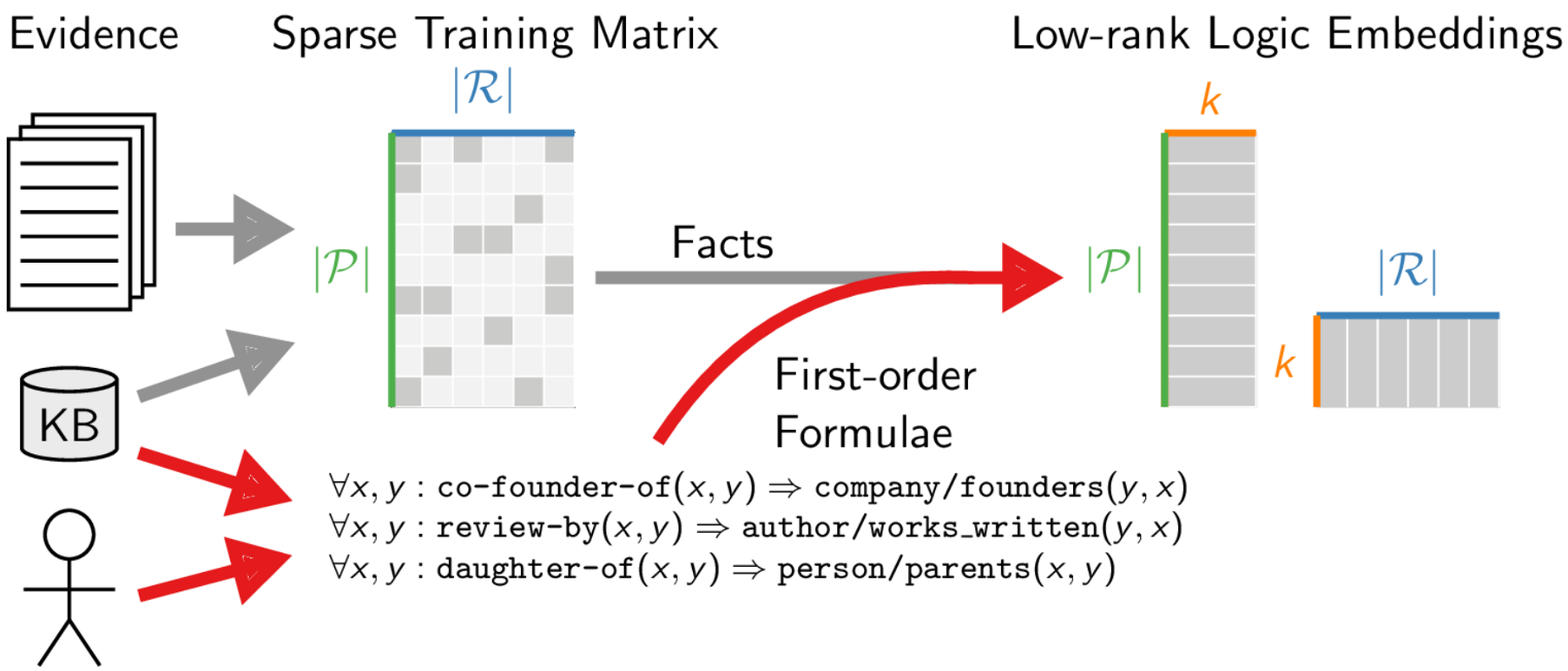
Overview



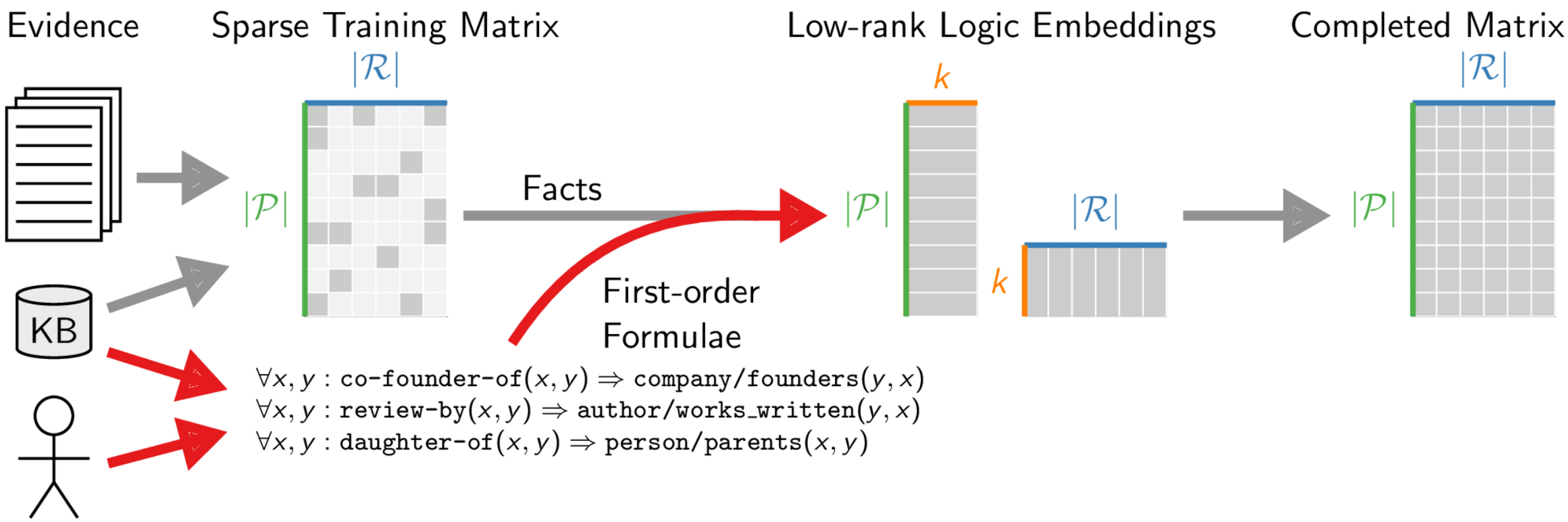
Overview


Evidence
Sparse Training Matrix
$|\mathcal{R}|$
$|\mathcal{P}|$
KB
$\forall x, y$: co-founder-of$(x, y) \Rightarrow$ company/founders(y, x)
$\forall x, y$: review-by$(x, y) \Rightarrow$ author/works_written(y, x)
$\forall x, y$: daughter-of$(x, y) \Rightarrow$ person/parents(x, y)

Overview


Evidence
Sparse Training Matrix
Low-rank Logic Embeddings
$|\mathcal{R}|$
$|\mathcal{P}|$
Facts
k
$|\mathcal{P}|$
$|\mathcal{R}|$
k
KB
First-order
Formulae
$\forall x, y : \texttt{co-founder-of}(x, y) \Rightarrow \texttt{company/founders}(y, x)$
$\forall x, y : \texttt{review-by}(x, y) \Rightarrow \texttt{author/works\_written}(y, x)$
$\forall x, y : \texttt{daughter-of}(x, y) \Rightarrow \texttt{person/parents}(x, y)$

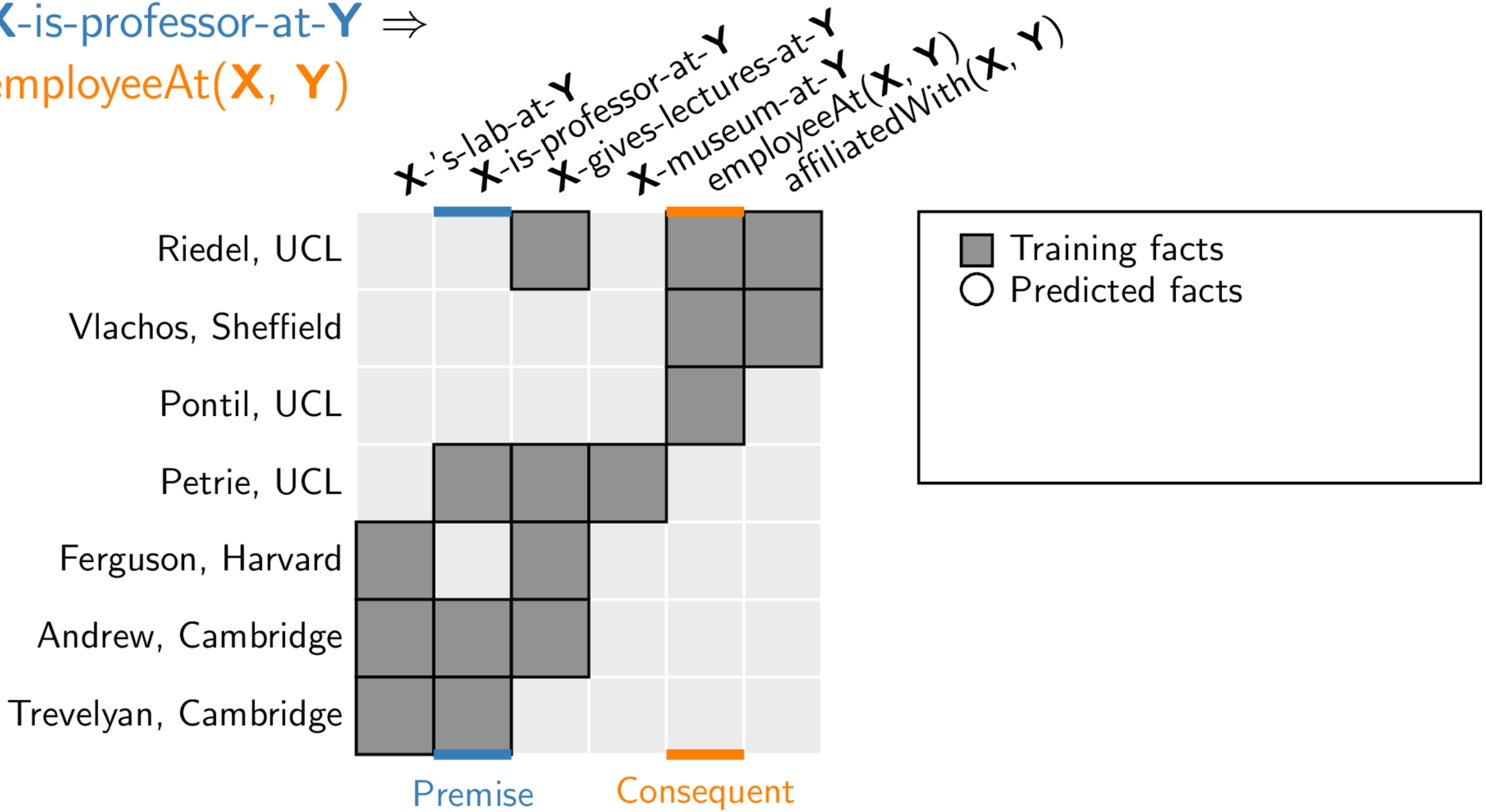
Overview


Evidence
Sparse Training Matrix
Low-rank Logic Embeddings
Completed Matrix
$|\mathcal{R}|$
$|\mathcal{P}|$
Facts
KB
First-order
Formulae
k
$\forall x, y : \texttt{co-founder-of}(x, y) \Rightarrow \texttt{company/founders}(y, x)$
$\forall x, y : \texttt{review-by}(x, y) \Rightarrow \texttt{author/works\_written}(y, x)$
$\forall x, y : \texttt{daughter-of}(x, y) \Rightarrow \texttt{person/parents}(x, y)$

Ways to combine matrix factorization and logic

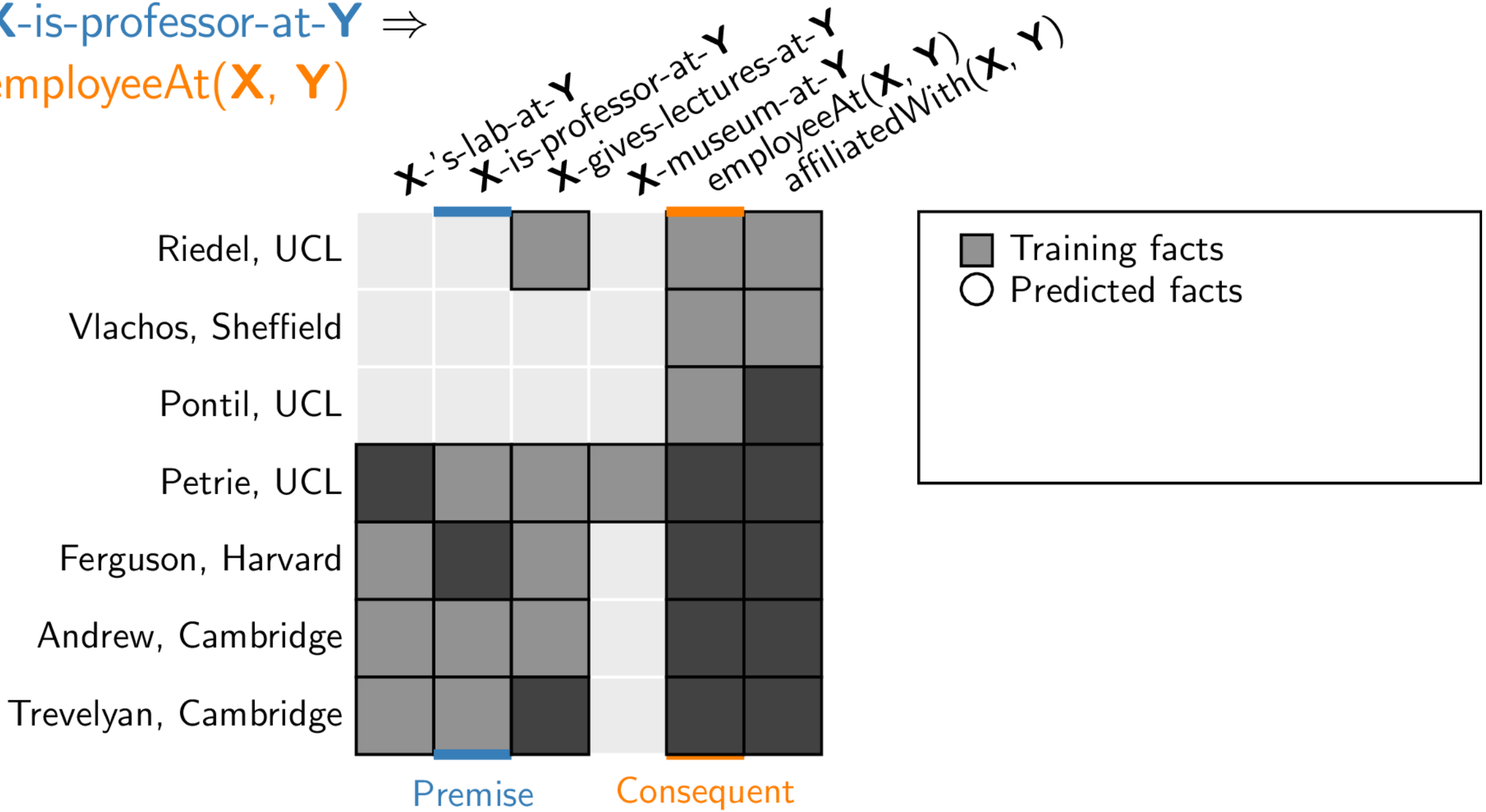
Ways to combine matrix factorization and logic

X-is-professor-at-**Y** $\Rightarrow$
employeeAt(**X**, **Y**)



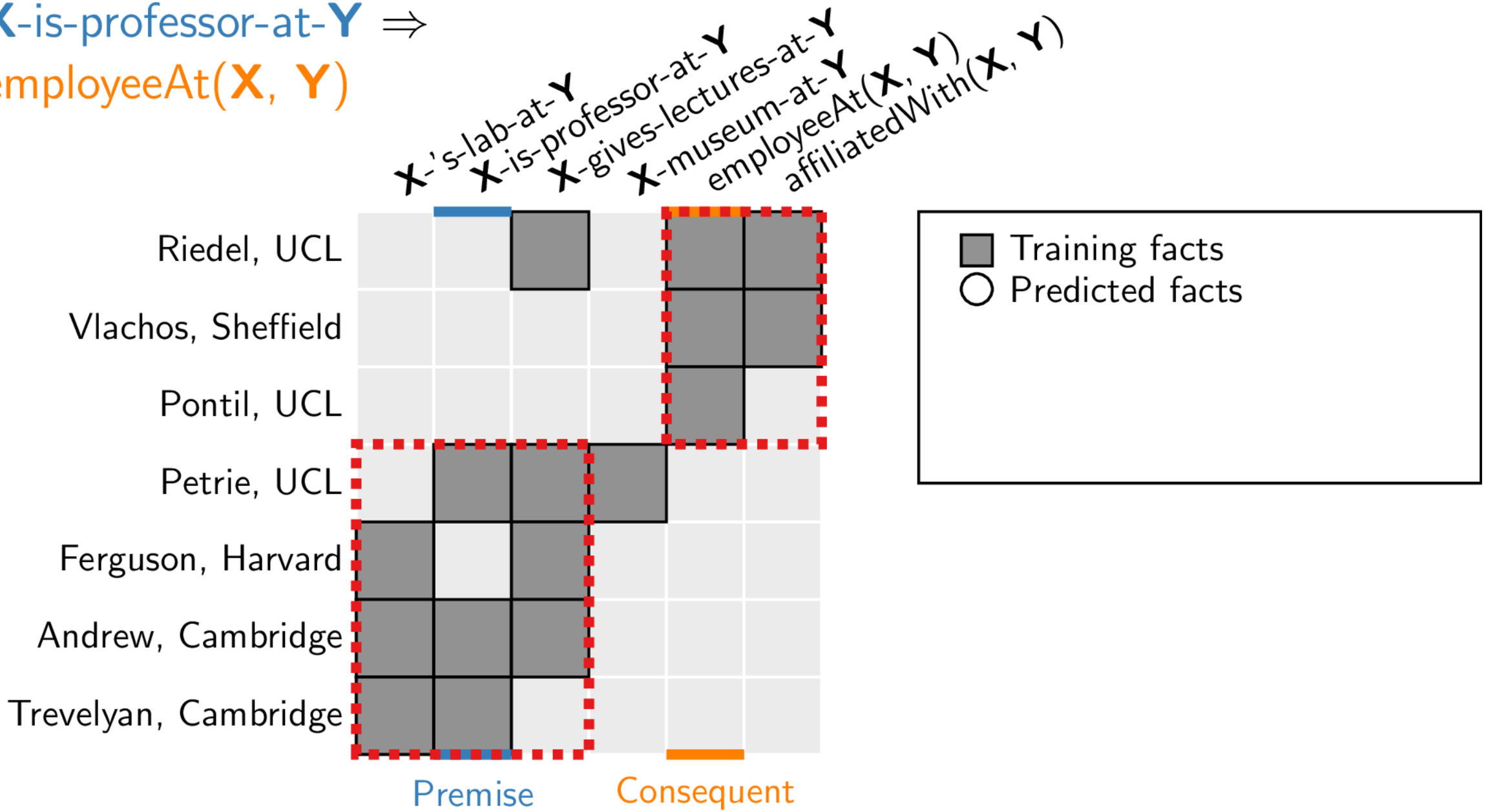
Ways to combine matrix factorization and logic

X-is-professor-at-**Y** ⇒
employeeAt(**X**, **Y**)



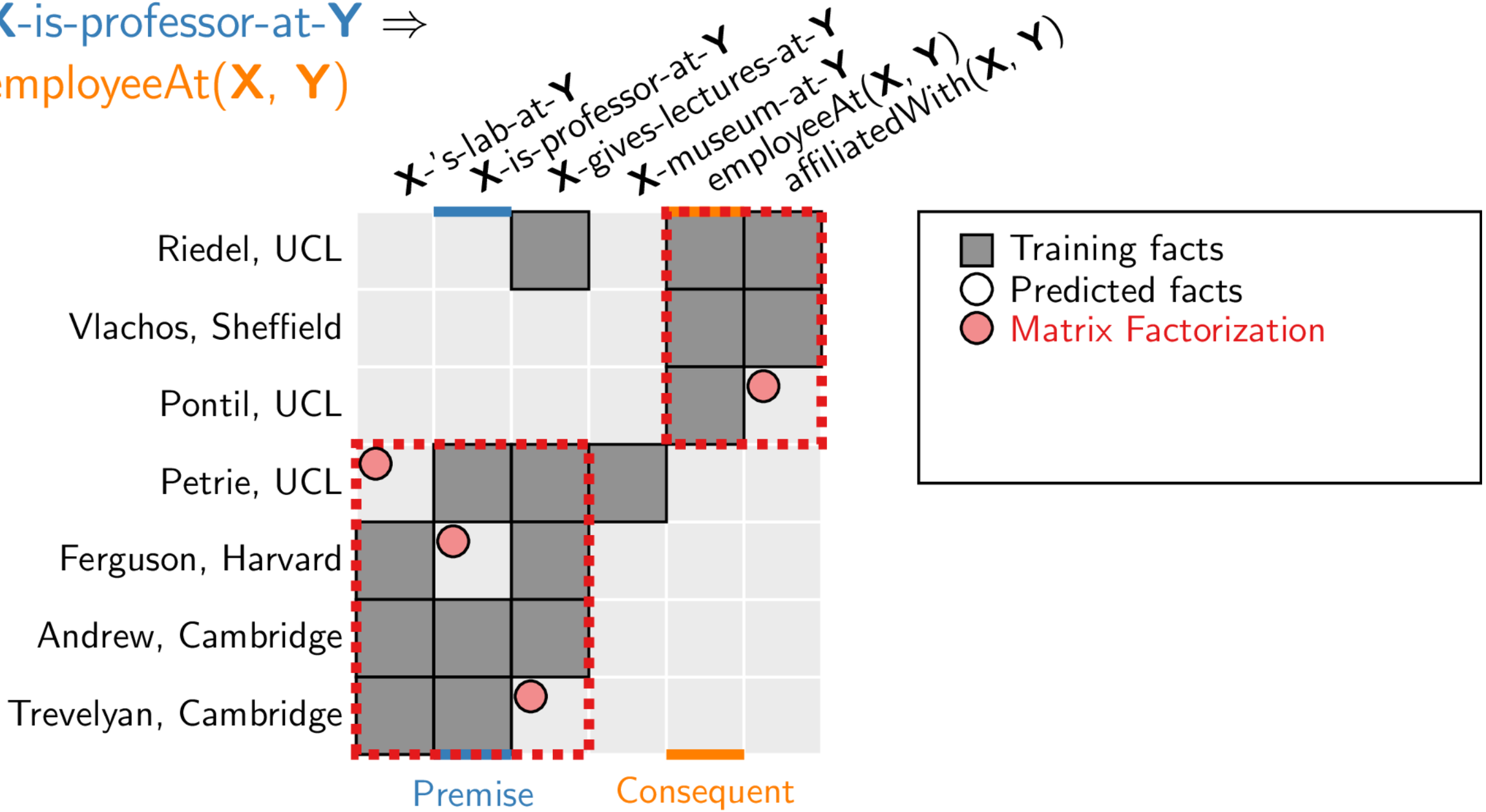
Ways to combine matrix factorization and logic

X-is-professor-at-**Y** $\Rightarrow$
employeeAt(**X**, **Y**)



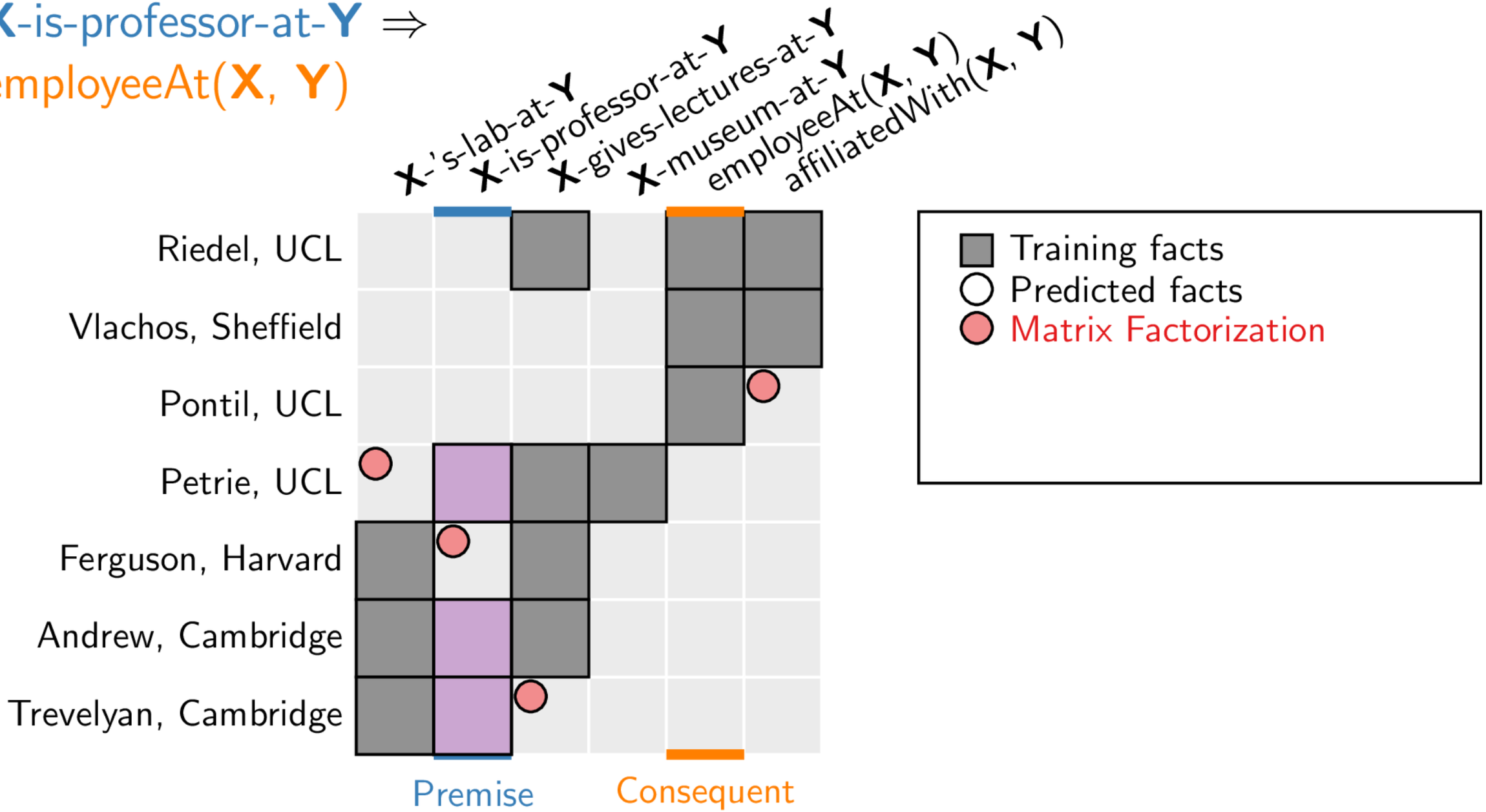
Ways to combine matrix factorization and logic

X-is-professor-at-**Y** $\Rightarrow$
employeeAt(**X**, **Y**)



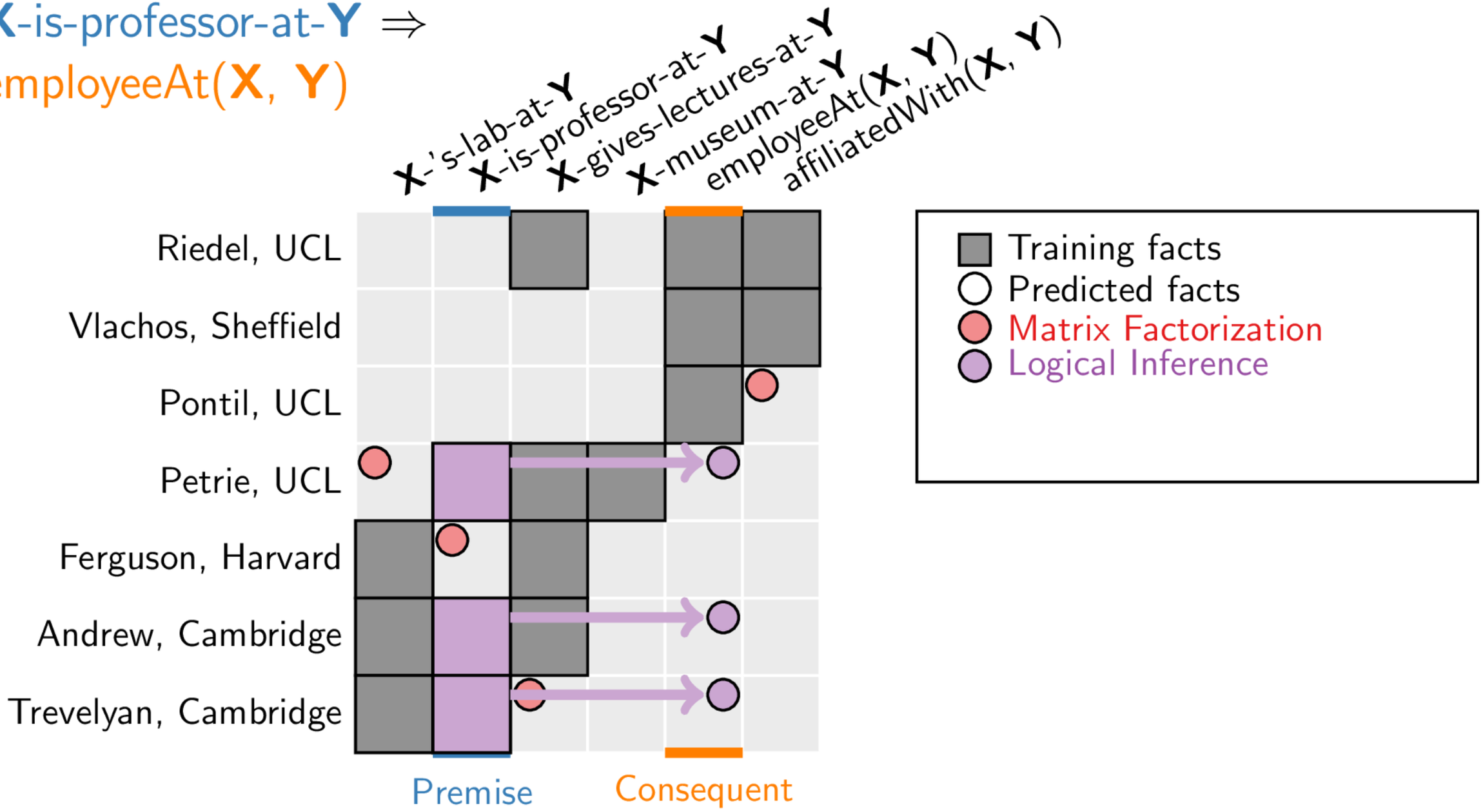
Ways to combine matrix factorization and logic

X-is-professor-at-**Y** $\Rightarrow$
employeeAt(**X**, **Y**)



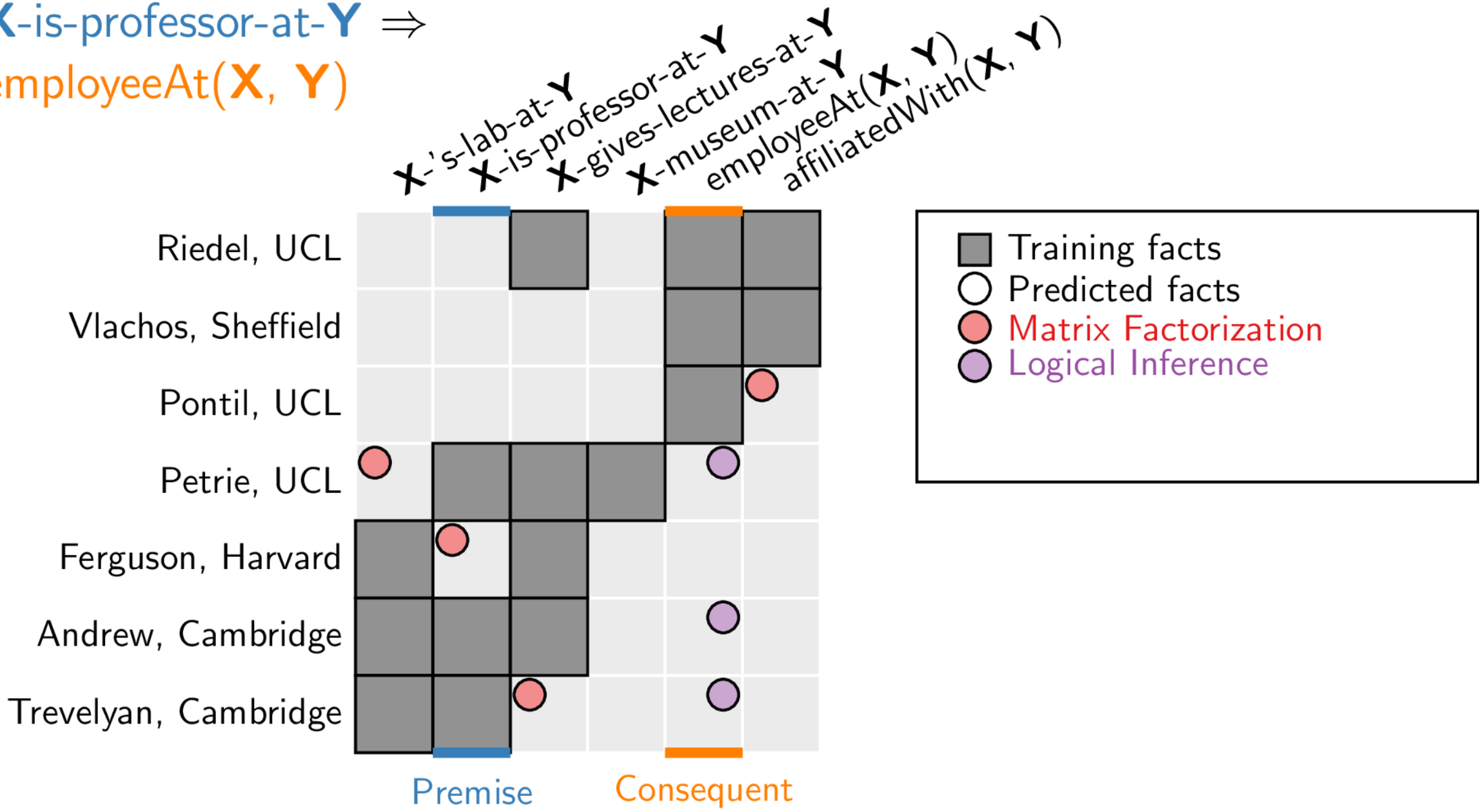
Ways to combine matrix factorization and logic

X-is-professor-at-**Y** $\Rightarrow$
employeeAt(**X**, **Y**)



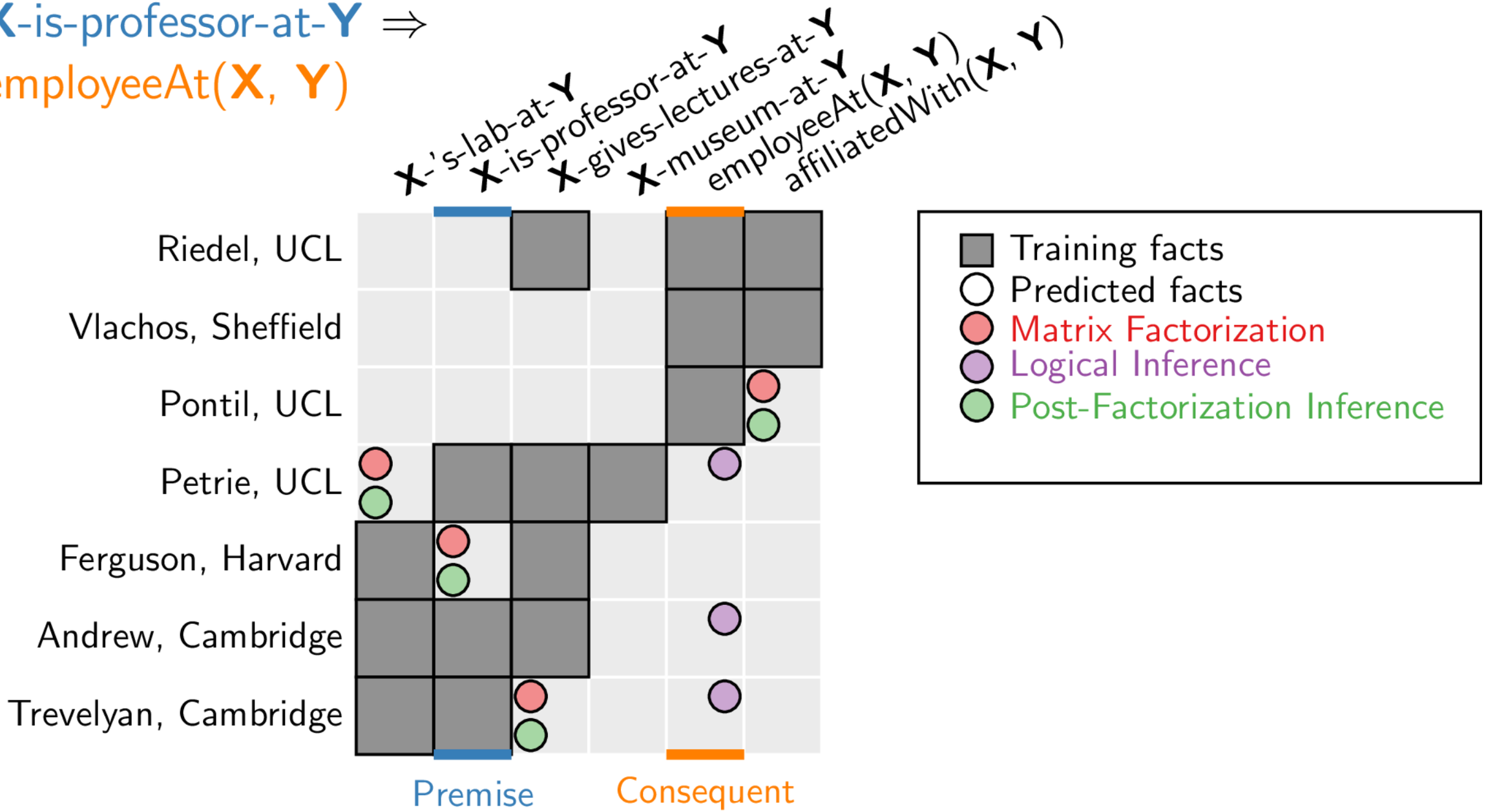
Ways to combine matrix factorization and logic

X-is-professor-at-**Y** ⇒
employeeAt(**X**, **Y**)



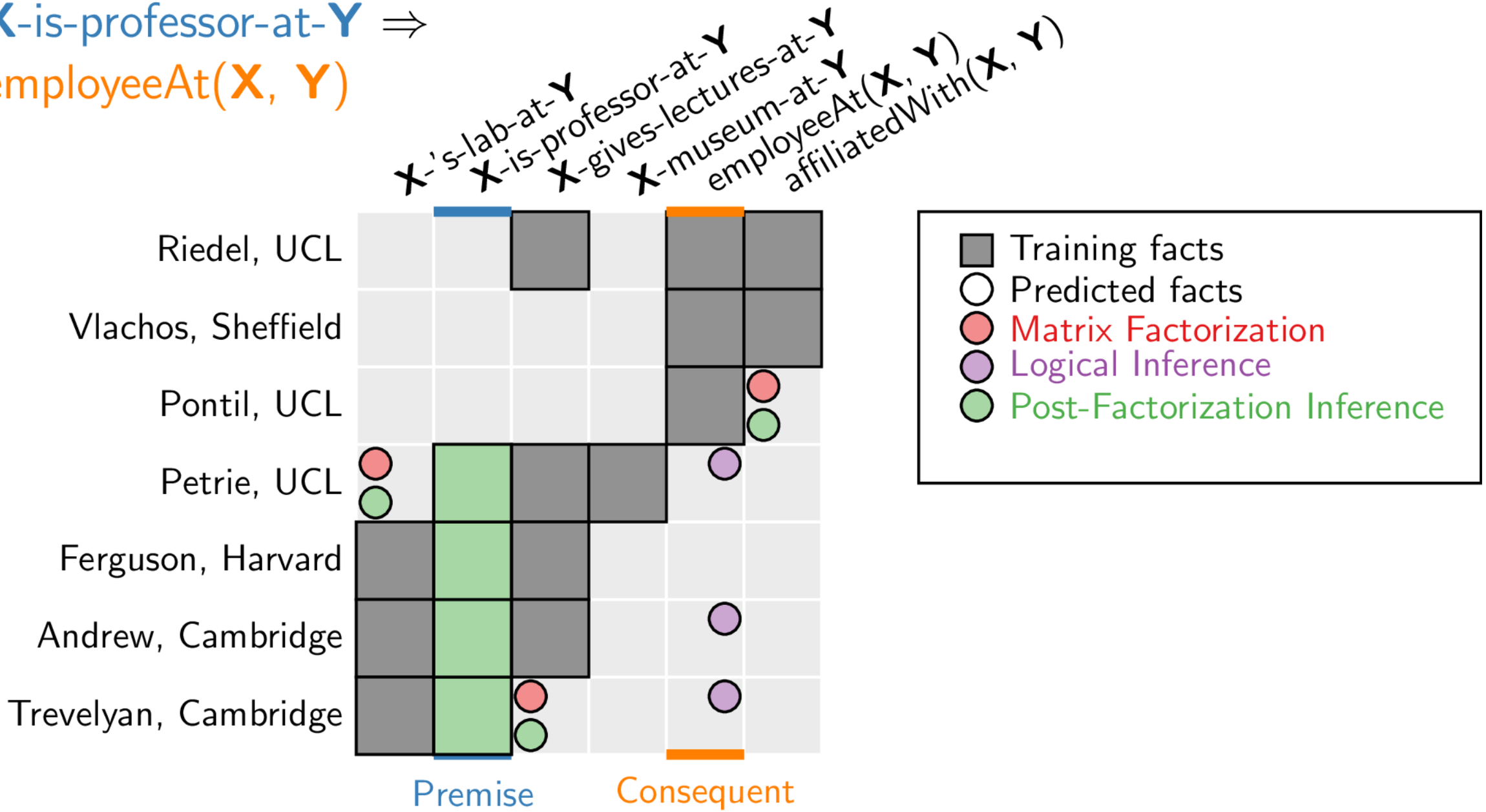
Ways to combine matrix factorization and logic

X-is-professor-at-**Y** ⇒
employeeAt(**X**, **Y**)



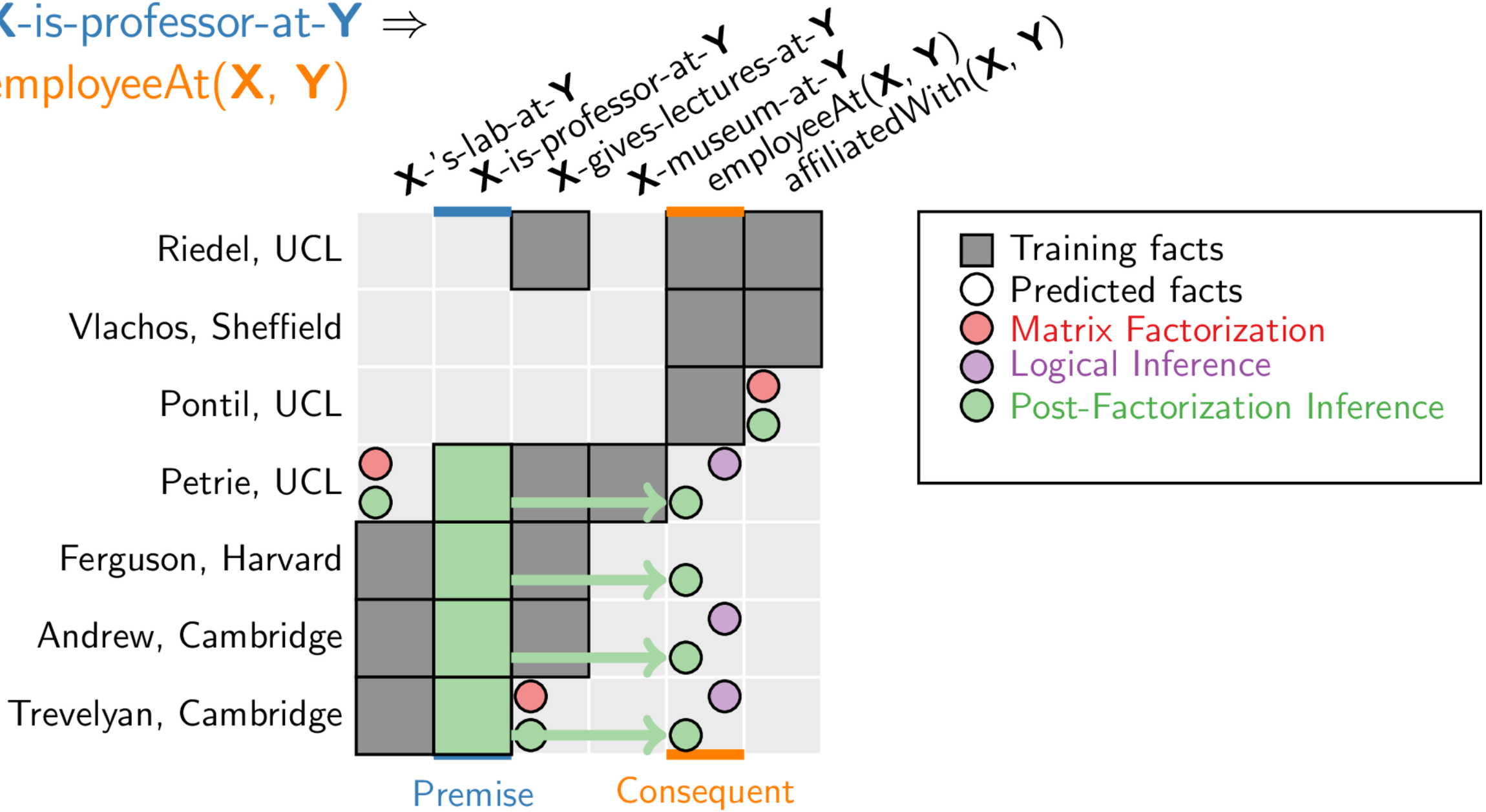
Ways to combine matrix factorization and logic

X-is-professor-at-**Y** ⇒
employeeAt(**X**, **Y**)



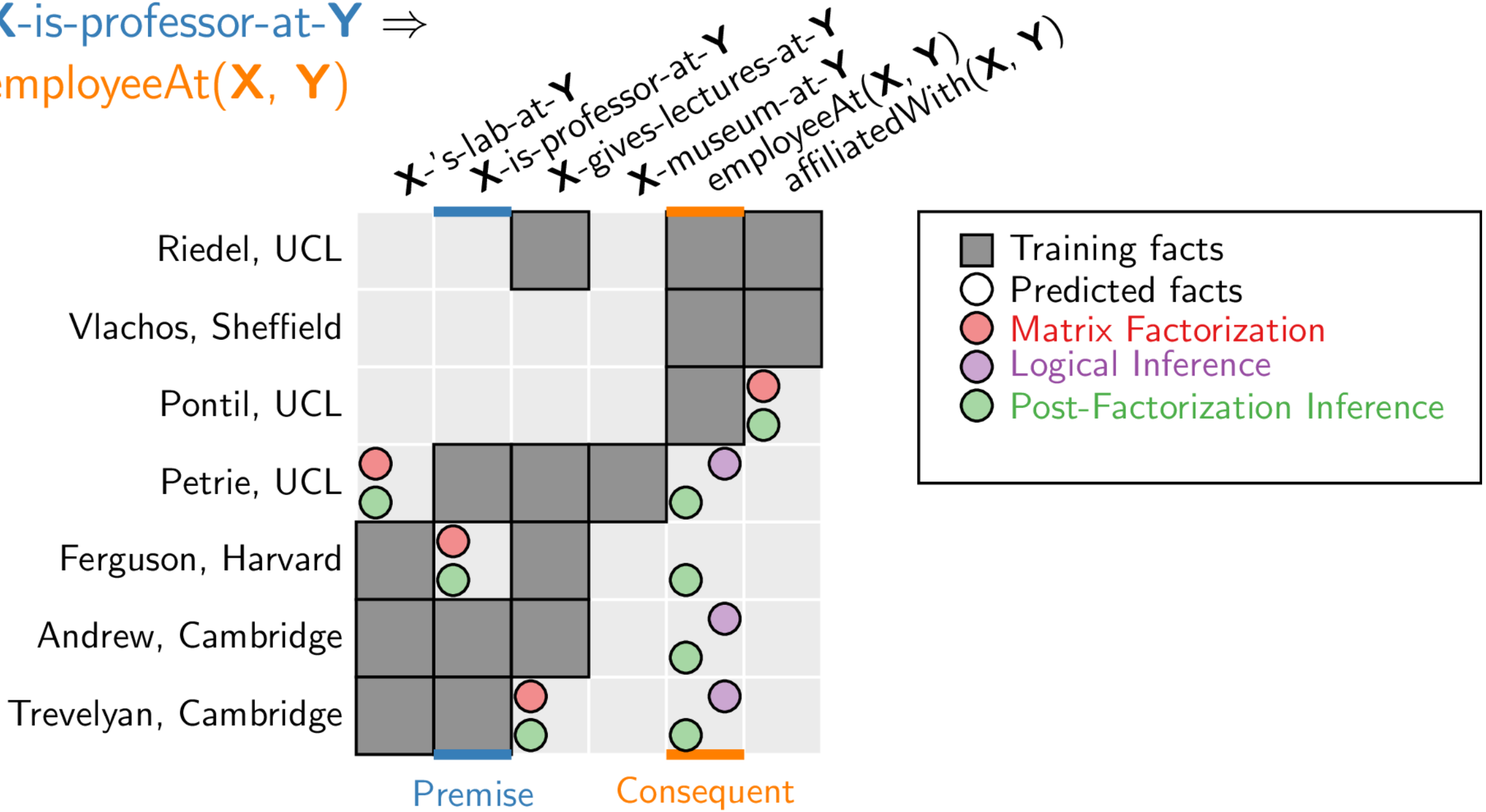
Ways to combine matrix factorization and logic

X-is-professor-at-**Y** ⇒
employeeAt(**X**, **Y**)



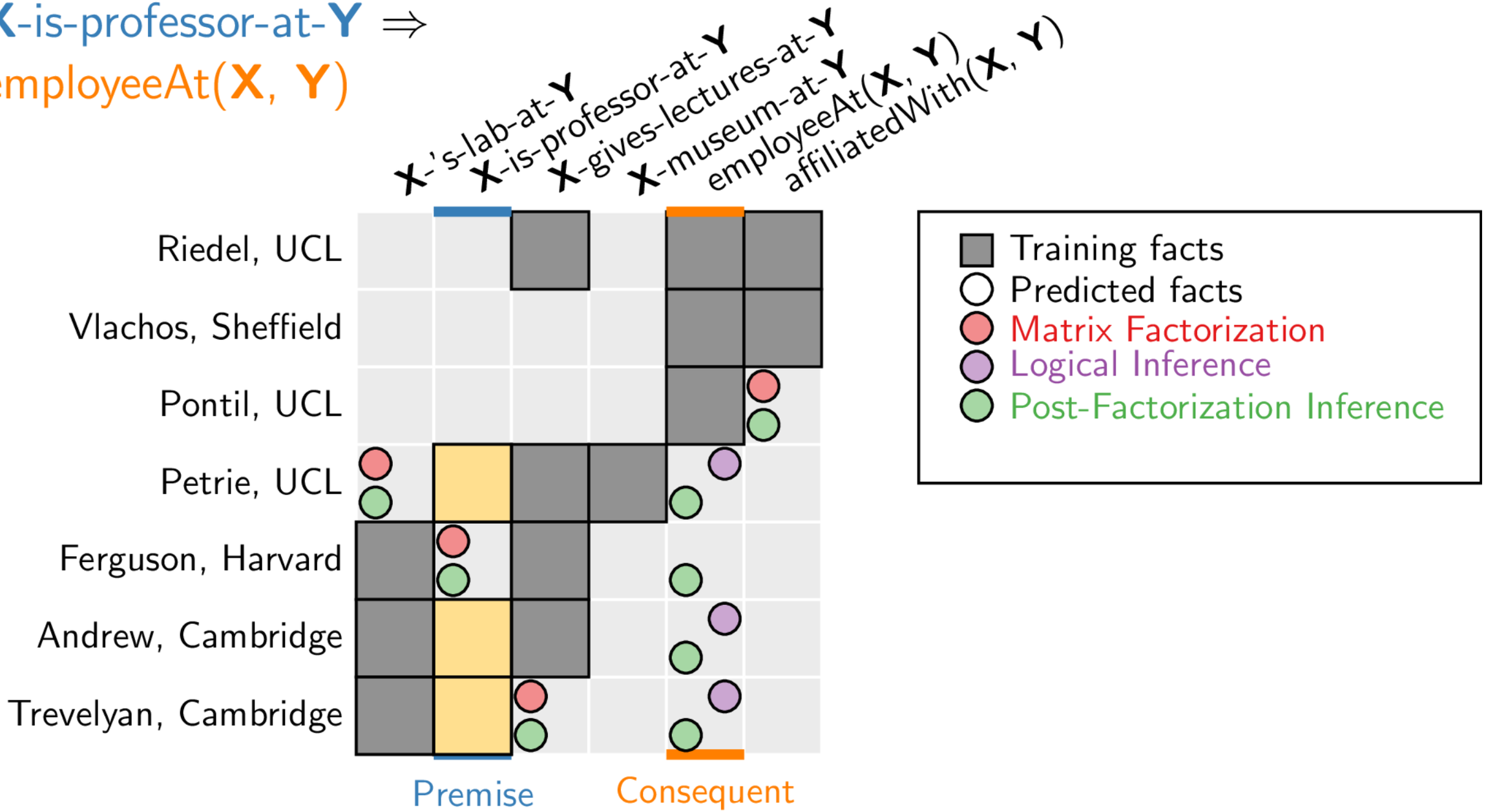
Ways to combine matrix factorization and logic

X-is-professor-at-**Y** ⇒
employeeAt(**X**, **Y**)



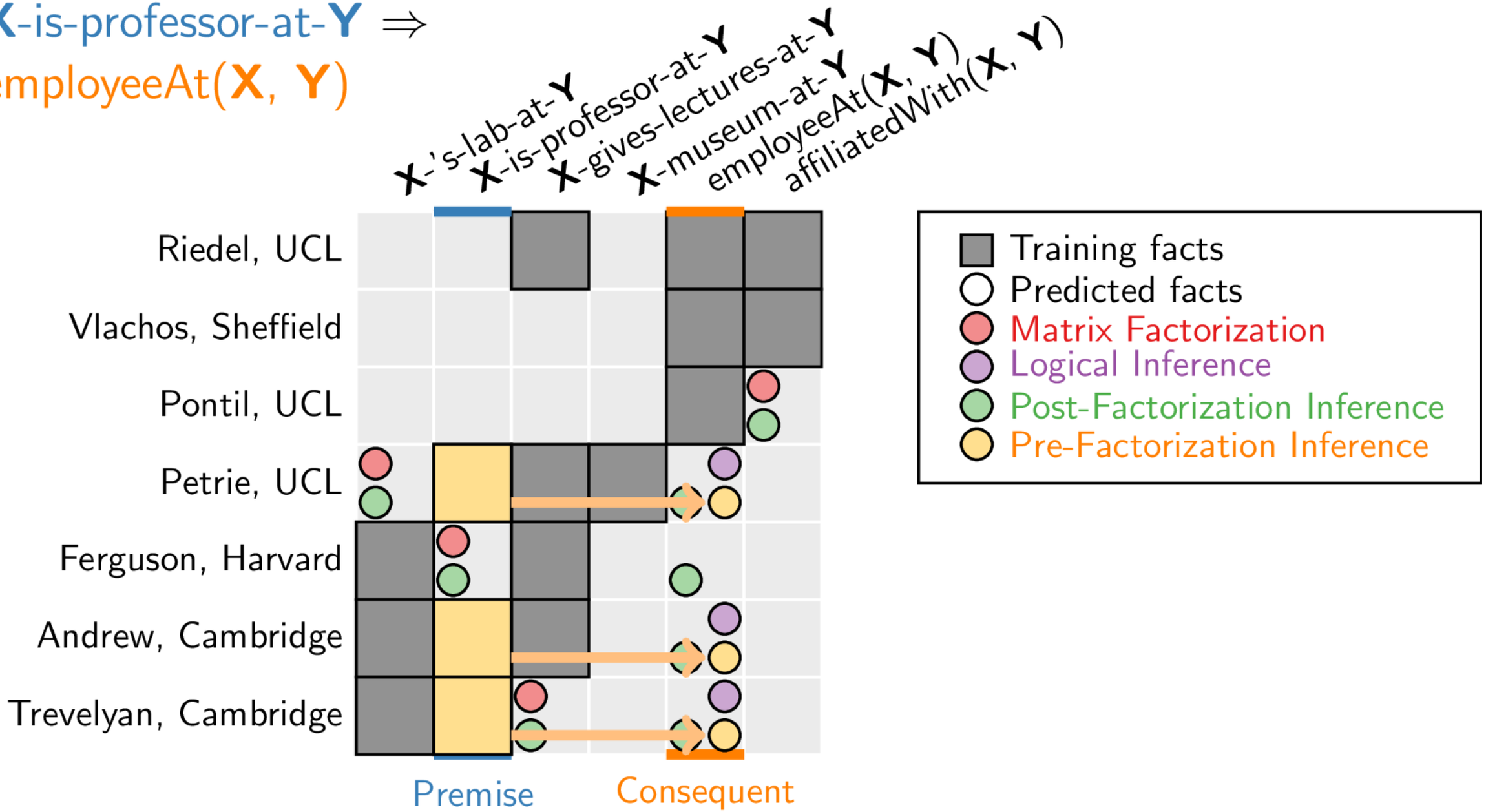
Ways to combine matrix factorization and logic

X-is-professor-at-**Y** ⇒
employeeAt(**X**, **Y**)



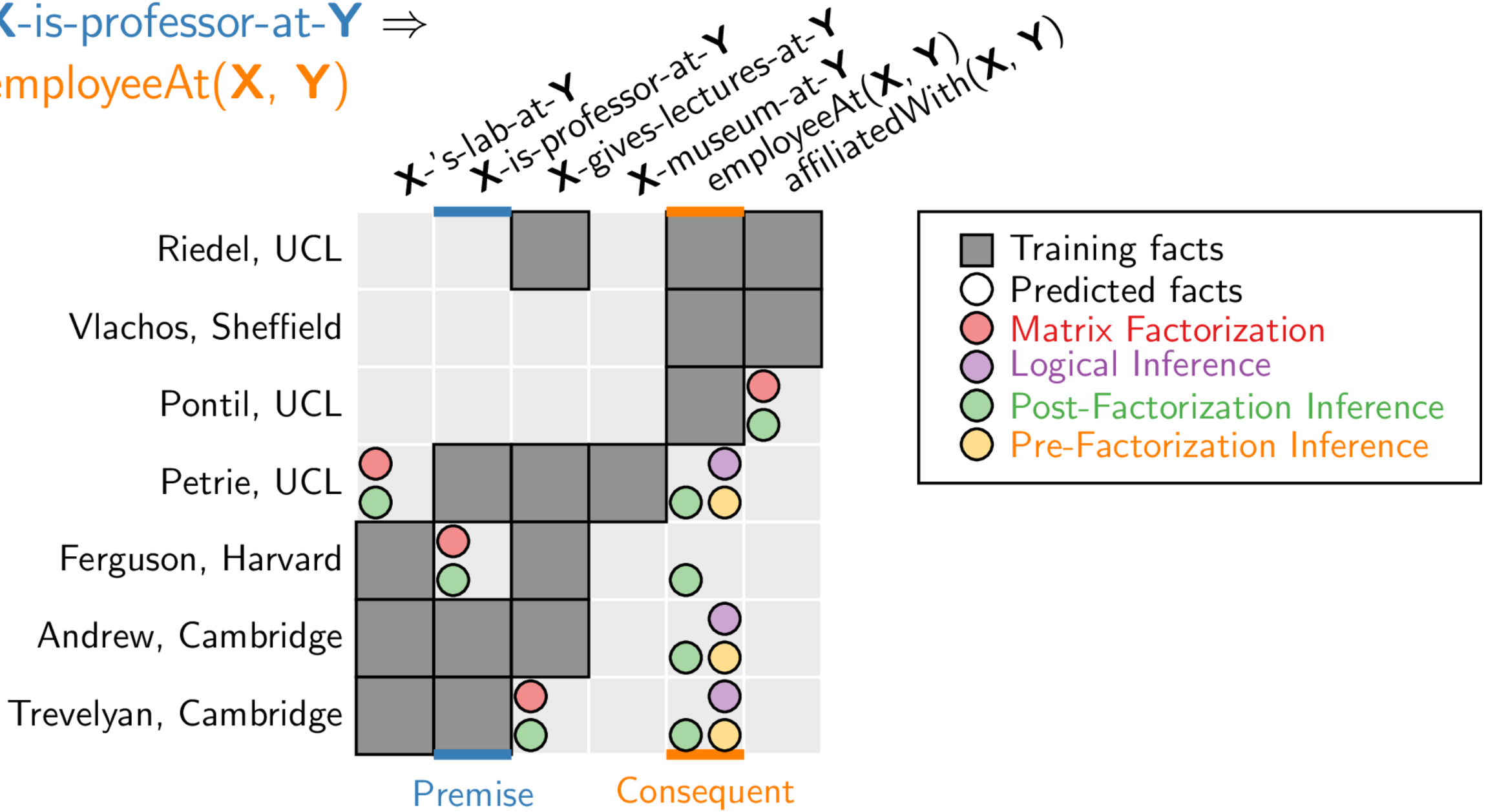
Ways to combine matrix factorization and logic

X-is-professor-at-**Y** ⇒
employeeAt(**X**, **Y**)



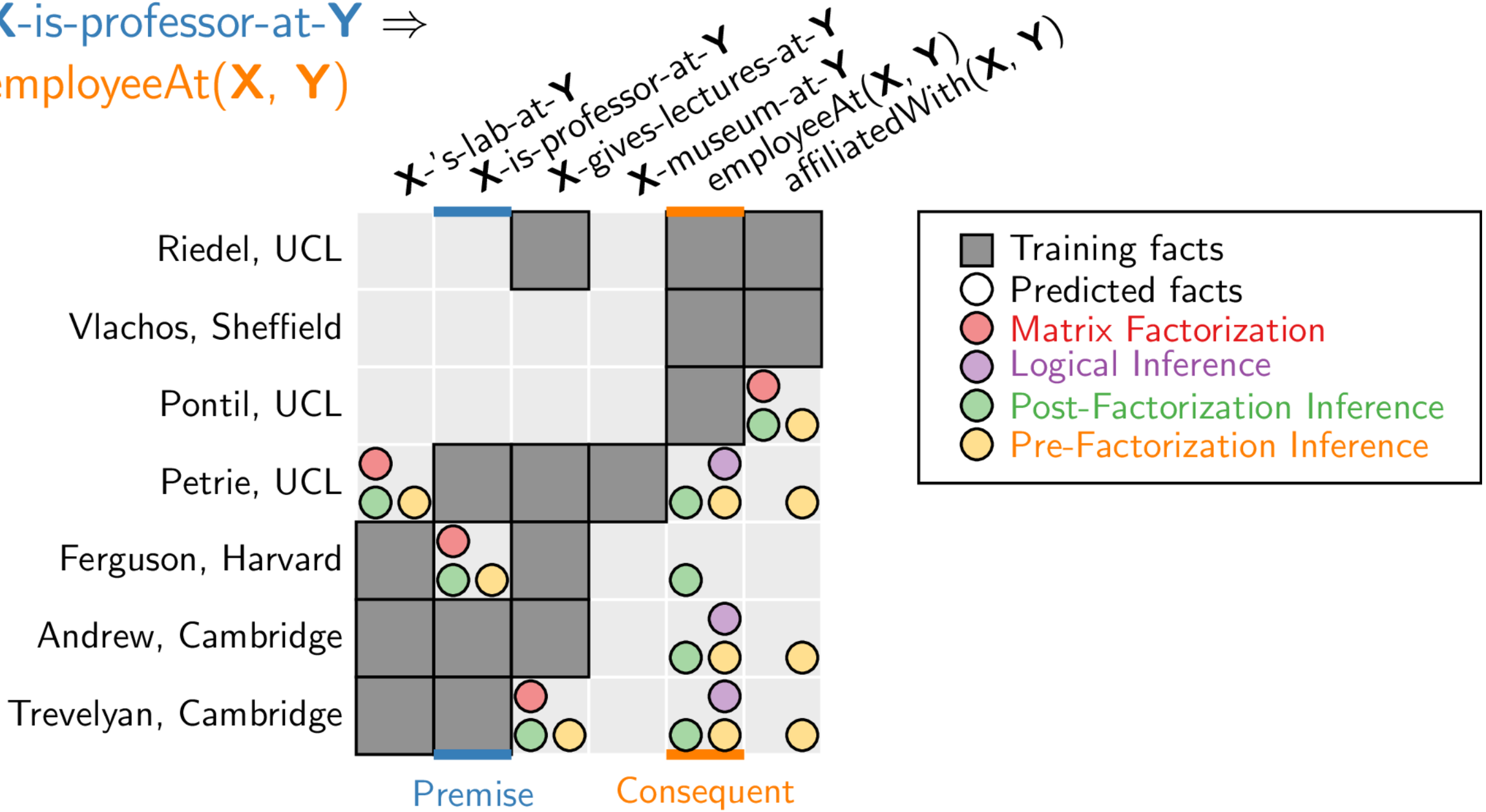
Ways to combine matrix factorization and logic

X-is-professor-at-**Y** ⇒
employeeAt(**X**, **Y**)

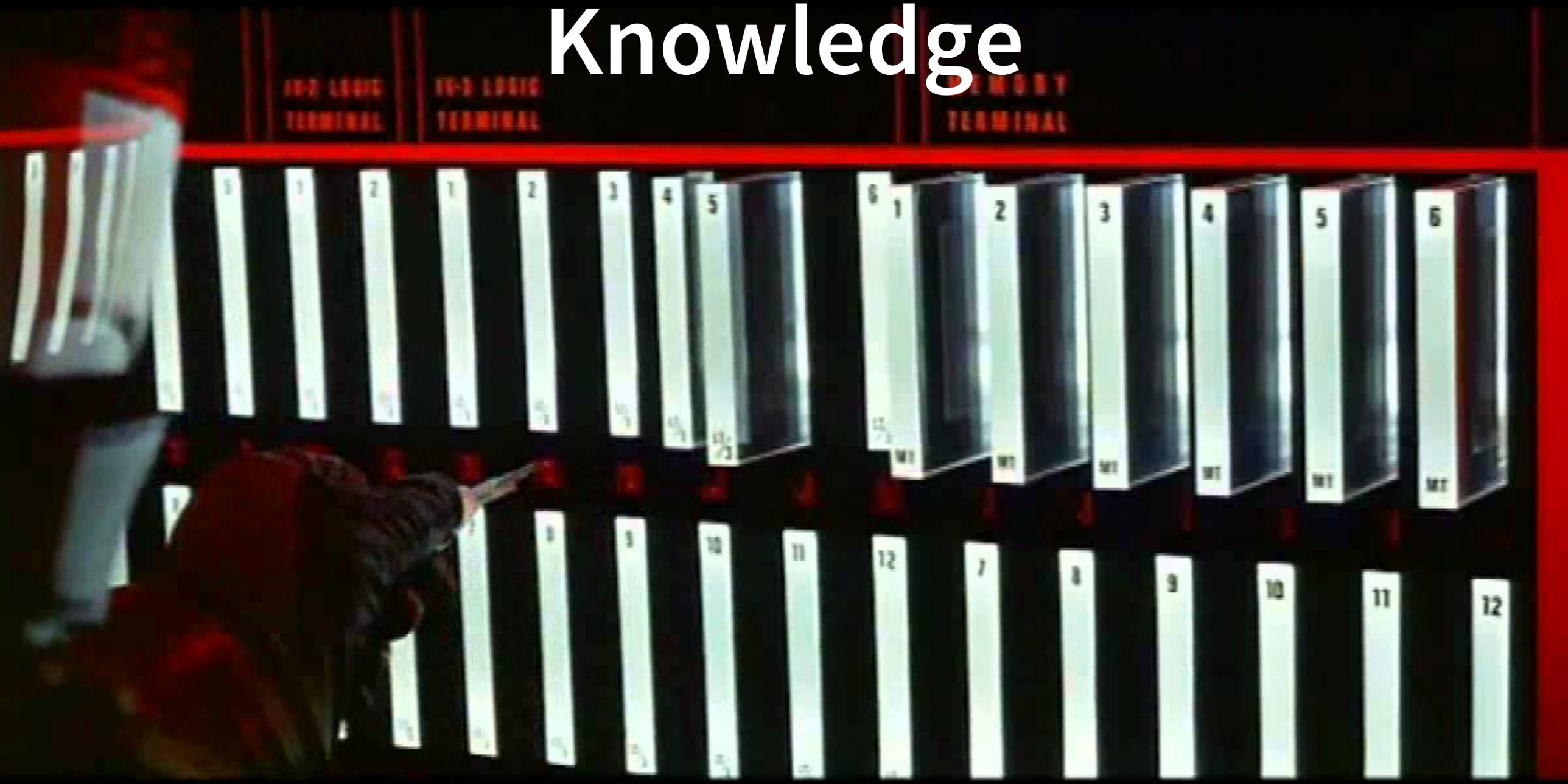


Ways to combine matrix factorization and logic

X-is-professor-at-**Y** ⇒
employeeAt(**X**, **Y**)



Injecting Logical Background Knowledge



Matrix factorization training objective

- Probability of $r_s(e_i, e_j)$ being true

$$p(r_s(e_i, e_j) = \text{true}) := \sigma(\mathbf{v}_s \cdot \mathbf{v}_{ij})$$

- Log-likelihood loss

$$\max_{\{\mathbf{v}_s\},\{\mathbf{v}_{ij}\}} \sum_{(s,i,j)\in\mathcal{T}} \log\left(\sigma(\mathbf{v}_s \cdot \mathbf{v}_{ij})\right) + \sum_{(s,u,w)\notin\mathcal{T}} \tau_{suw} \log\left(1 - \sigma(\mathbf{v}_s \cdot \mathbf{v}_{uw})\right)$$

- Optimized using AdaGrad (Duchi et al., 2011)

Matrix factorization is "injecting" atomic formulae

- Training facts are ground atoms: $\mathcal{F} = r_s(e_i, e_j)$
- Let [ ] denote mapping from symbolic formulae to expectations
- $[r_s(e_i, e_j)] := \sigma(\mathbf{v}_s \cdot \mathbf{v}_{ij})$
- Training objective:

$$\max_{\{\mathbf{v}_s\},\{\mathbf{v}_{ij}\}} \sum_{\mathcal{F}\in\mathfrak{F}} \log([\mathcal{F}])$$

- Can we do this for any propositional formulae?

$$[\mathcal{F}] = [r_1(e_i\ e_j) \wedge \neg r_2(e_i\ e_j) \Rightarrow r_3(e_i\ e_j)]$$

Differentiable logic formulae

$$[\mathcal{F}] = \begin{cases} \sigma(\mathbf{v}_s \cdot \mathbf{v}_{ij}) & \text{if } \mathcal{F} = r_s(e_i, e_j), \text{ i.e., facts} \\ 1 - [\mathcal{A}] & \text{if } \mathcal{F} = \neg\mathcal{A} \\ [\mathcal{A}] * [\mathcal{B}] & \text{if } \mathcal{F} = \mathcal{A} \wedge \mathcal{B} \end{cases}$$

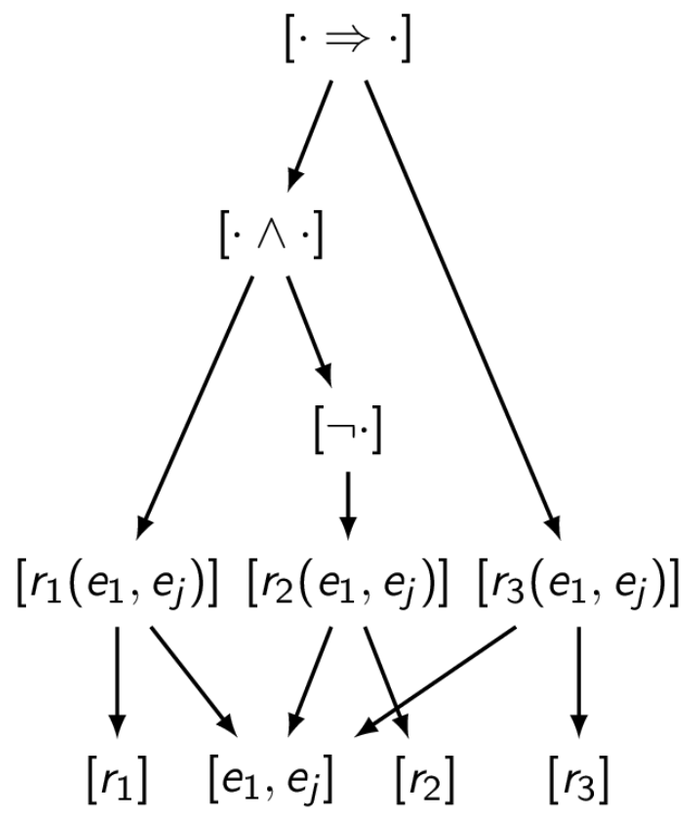
- From negation and conjunction we can build any propositional formula
  - $[\mathcal{A} \vee \mathcal{B}] = 1 - (1 - [\mathcal{A}]) * (1 - [\mathcal{B}])$
  - $[\mathcal{A} \Rightarrow \mathcal{B}] = [\mathcal{A}]([\mathcal{B}] - 1) + 1$
- Jointly maximize the log-likelihood of atomic and propositional formulae

$$\max_{\{\mathbf{v}_s\},\{\mathbf{v}_{ij}\}} \sum_{\mathcal{F} \in \mathfrak{F}} \log([\mathcal{F}])$$

Backpropagation through structure

(Goller and Küchler, 1996)

$$r_1(e_i, e_j) \wedge \neg r_2(e_i, e_j) \Rightarrow r_3(e_i, e_j)$$



Grounding

$$\forall x, y : r_s(x, y) \Rightarrow r_t(x, y)$$

- Grounding based on all observed facts for premise $r_s(e_i, e_j)$ and consequent $r_t(e_i, e_j)$
- Sample unobserved facts $r_s(e'_i, e'_j)$ and $r_t(e'_i, e'_j)$
- Add propositional formulae to the matrix factorization training objective
  - $[r_s(e_i, e_j) \Rightarrow r_t(e_i, e_j)]$
  - $[r_s(e'_i, e'_j) \Rightarrow r_t(e'_i, e'_j)]$

Experiments

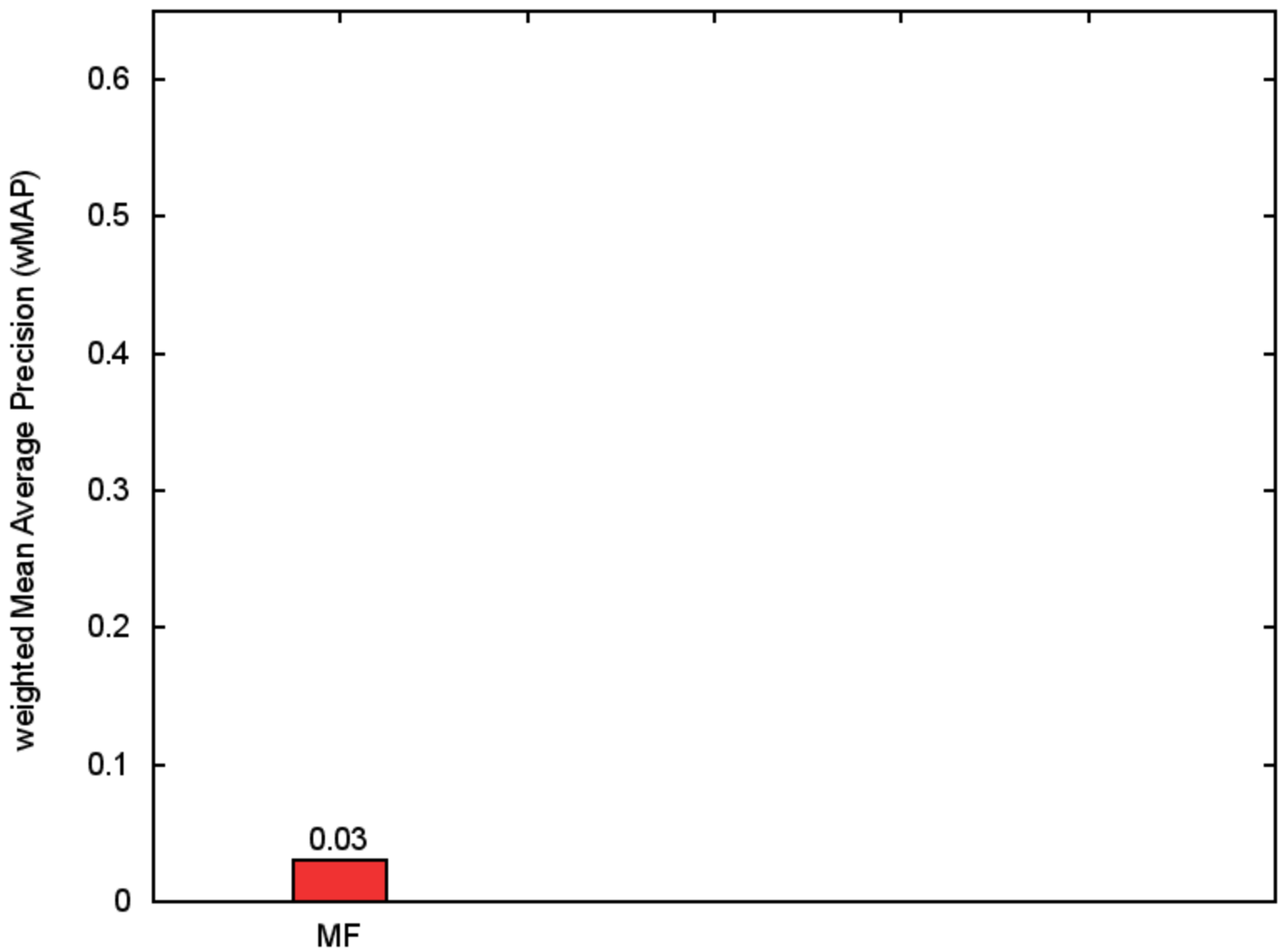
- Relation extraction corpus (Riedel et al., 2013)
  - ~4k textual patterns from New York Times corpus and 151 Freebase relations, ~42k entity-pairs, ~100k training facts
  - Metric: mean average precision (**MAP**) on manually annotated predictions for Freebase relations
- **Zero-shot Relation Learning**:
  - All Freebase training facts are removed
  - No alignment between Freebase relations and textual patterns
- **Relations with Few Distant Labels**:
  - Varying degree of alignment between relations and textual patterns

Evaluation

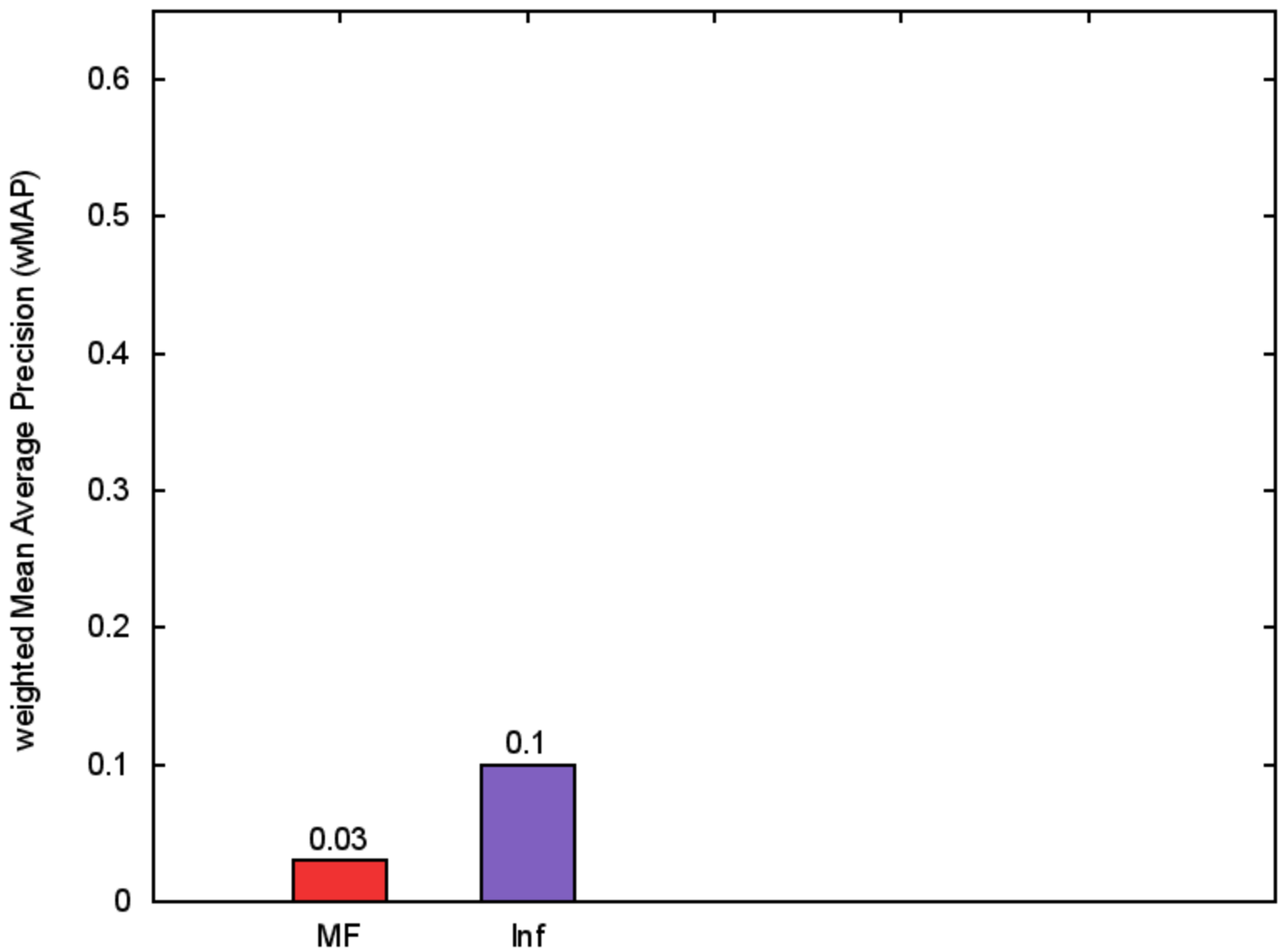
- Formulae extracted from predictions of a matrix factorization model (Sanchez et al., 2015)
- Annotated manually
- Given these formulae, which method can best make use of them?
  - **Logical Inference**
  - **Post-Factorization Inference**
  - **Pre-Factorization Inference**
  - **Joint Optimization**
- Example:
  $\forall x, y : \texttt{\#2-minister-\#1}(x, y) \Rightarrow \texttt{person/nationality}(x, y)$

Zero-shot relation learning

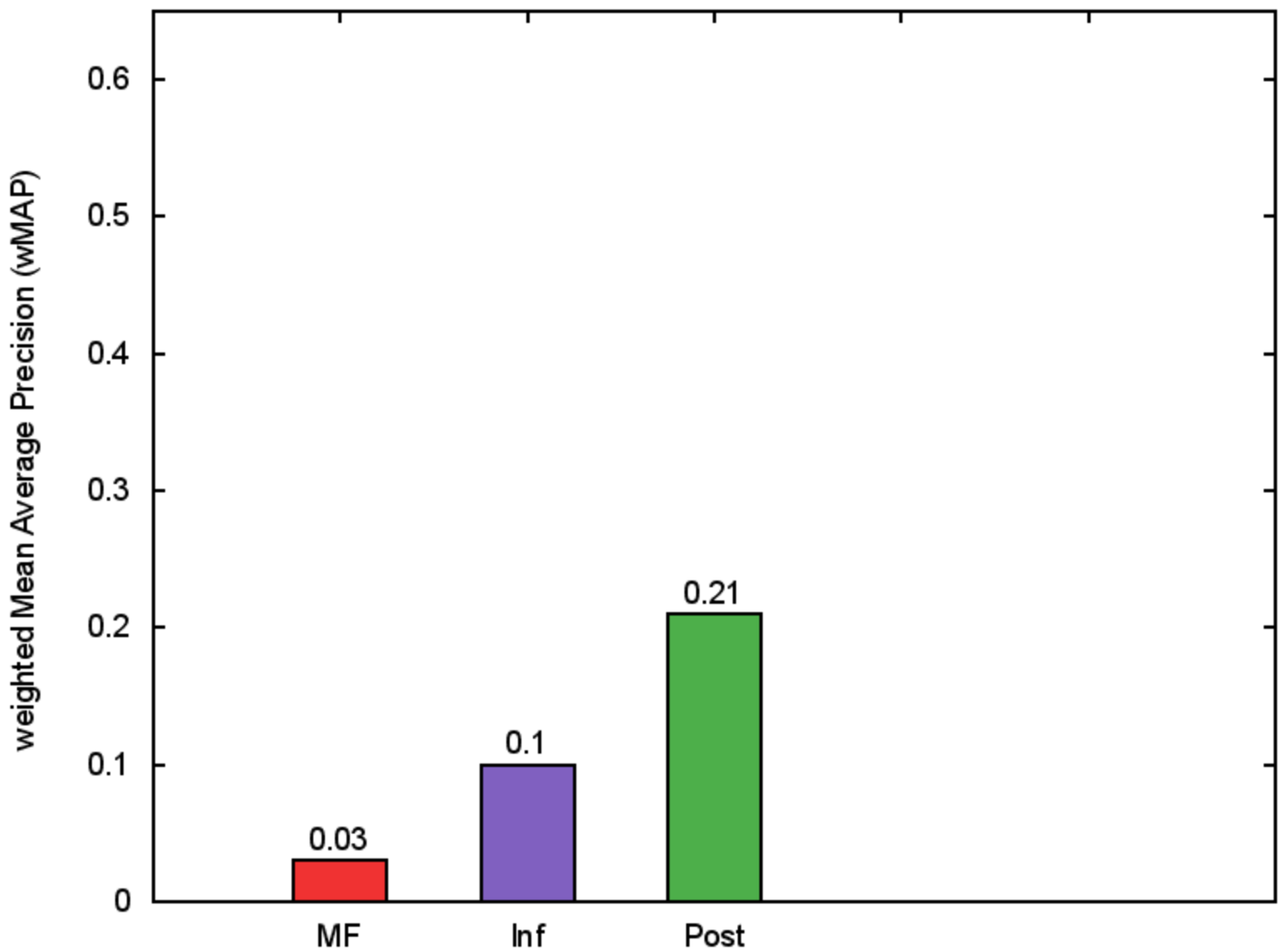
Zero-shot relation learning



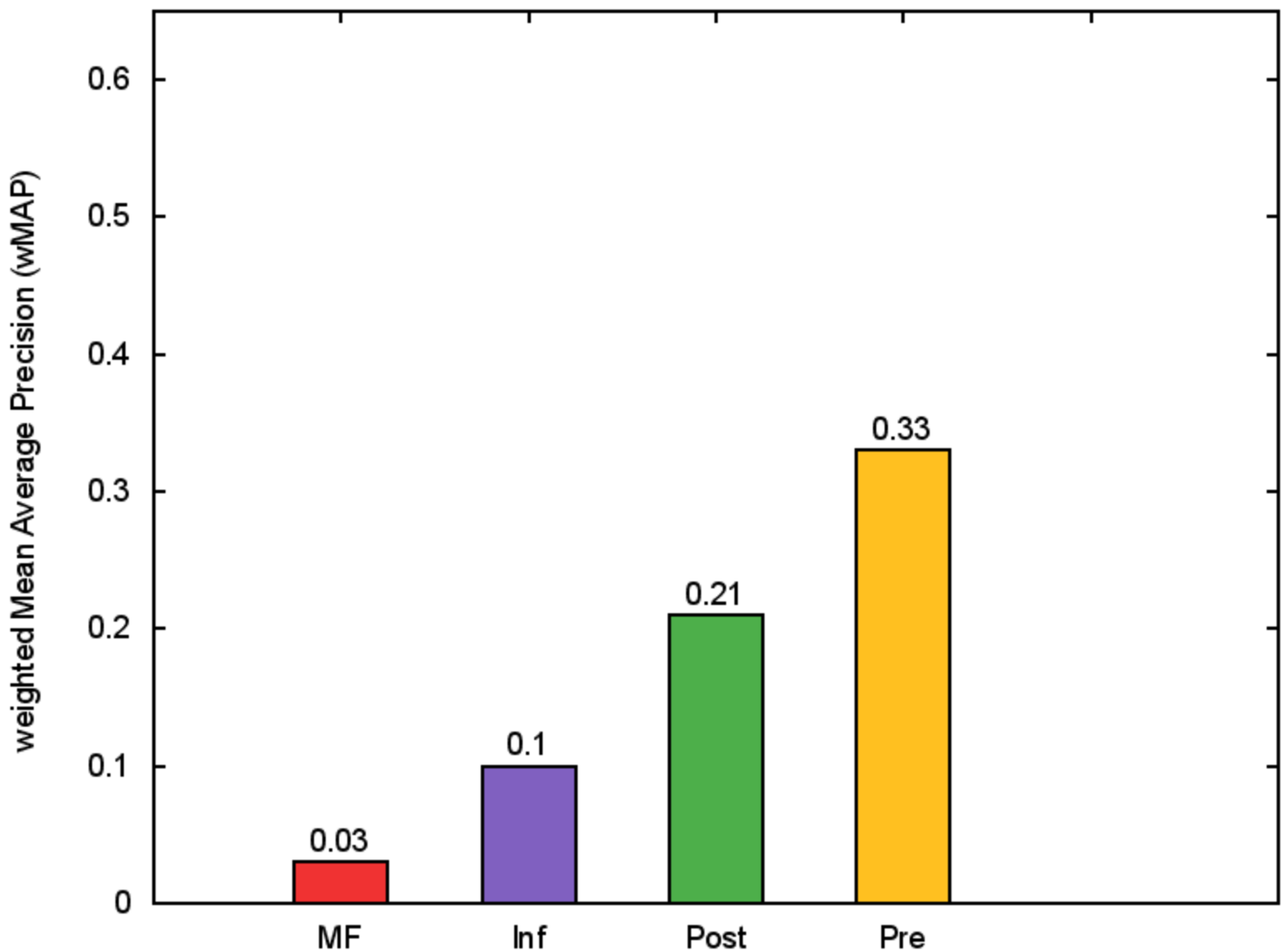
Zero-shot relation learning



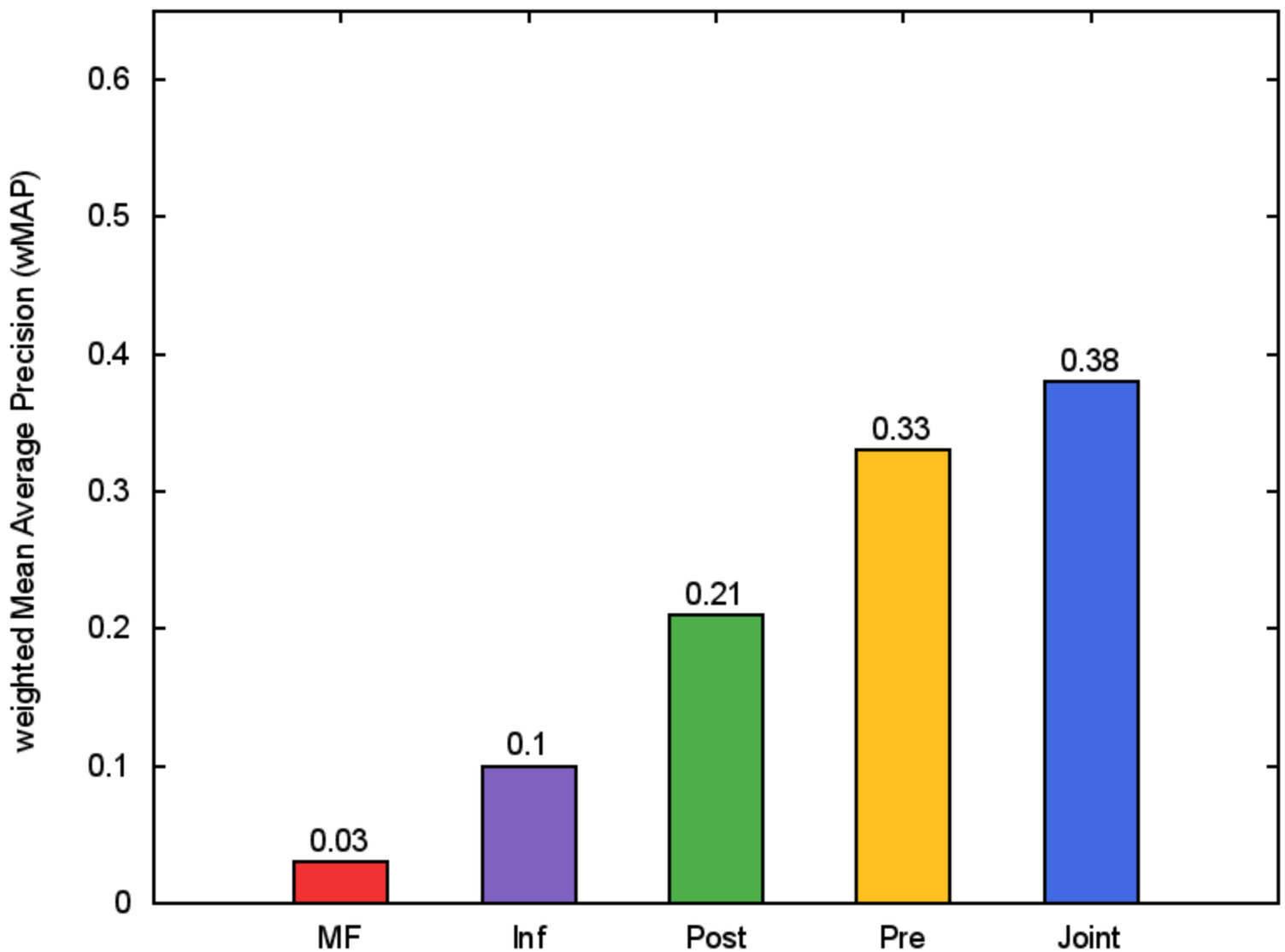
Zero-shot relation learning



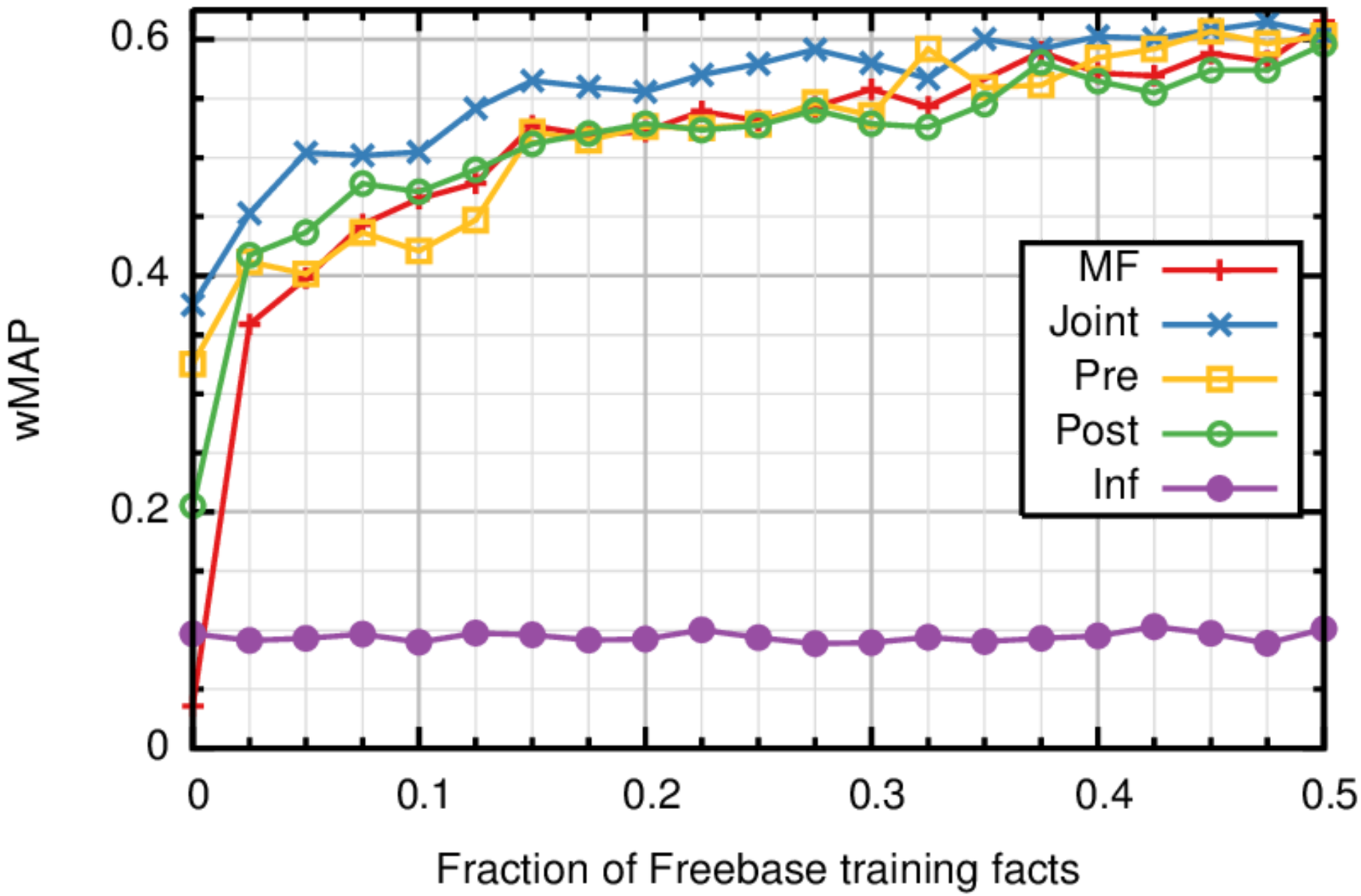
Zero-shot relation learning



Zero-shot relation learning



Relations with few distant labels



Summary

- Matrix factorization for relation extraction generalizes well
  - Hard to fix mistakes
  - Fails for new or spare relations
- Formalize background knowledge as logical formulae
  - Brittle and no generalization
- Formulae can be injected into embeddings
  - Make logical background knowledge differentiable
  - Jointly optimize over factual and first-order knowledge
  - **Learns relations with no or little prior information in databases**
  - **Generalizes beyond textual patterns mentioned in formulae**
  - **Joint optimization > Pre-factorization inference > Post-factorization inference > Logical inference**

Matrix Factorization in Wolfe

```
//training loss, stochastic term based on sampling a positive cell, and then a neg
def loss(t: Thetas.Term): DoubleTerm = {
  //we sample a positive cell, and memoize the result
  val pos = mem(trainingData.sampleShuffled)
  //based on the memoized positive cell, we sample a negative cell which needs to
  val neg = mem(sampleNegCell(pos))
  //the loss based on positive and negative cell
  log(sigm(score(t)(pos))) + log(sigm(-score(t)(neg))) + regularize(t)(pos, neg)
}

//the per cell score
def score(theta: Thetas.Term)(cell: Cells.Term): DoubleTerm =
  theta.rows(cell.row) dot theta.cols(cell.col)
```

We are giving a demo at 5pm today!

Thank you!

Questions?

rockt.github.com