Non-Parametric Path Based Model for Taxonomy Induction in Knowledge Graphs

Marcin Pietrasik, Marek Reformat, Anna Wilbik

Knowledge Graphs

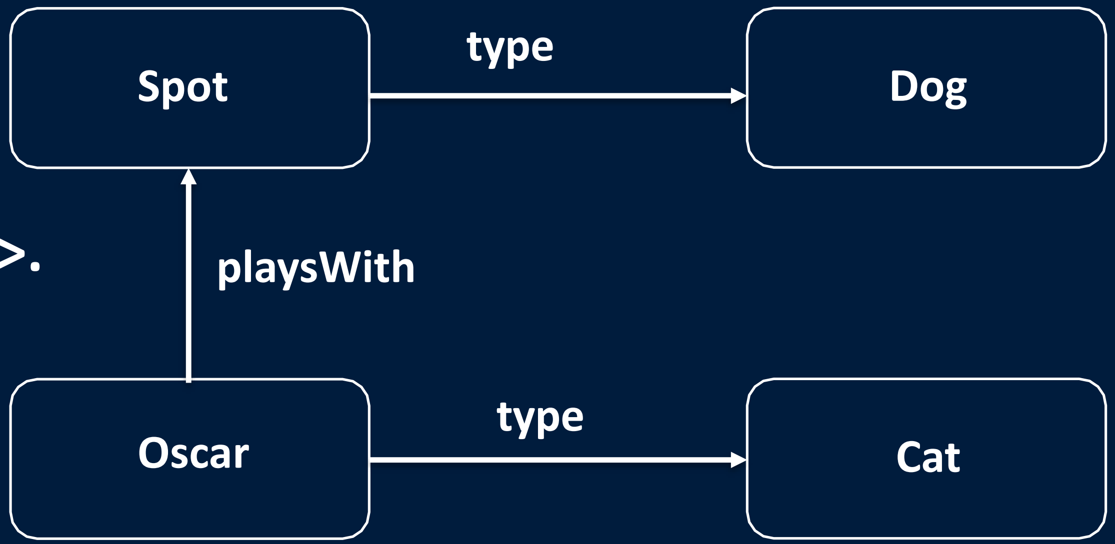
- Triples are how data is stored in a knowledge graph
  - Links a **subject** (head) to and **object** (tail) via a **predicate**

<Oscar> <type> <Cat>.

Knowledge Graphs

- Put triples together and you get a **knowledge graph**

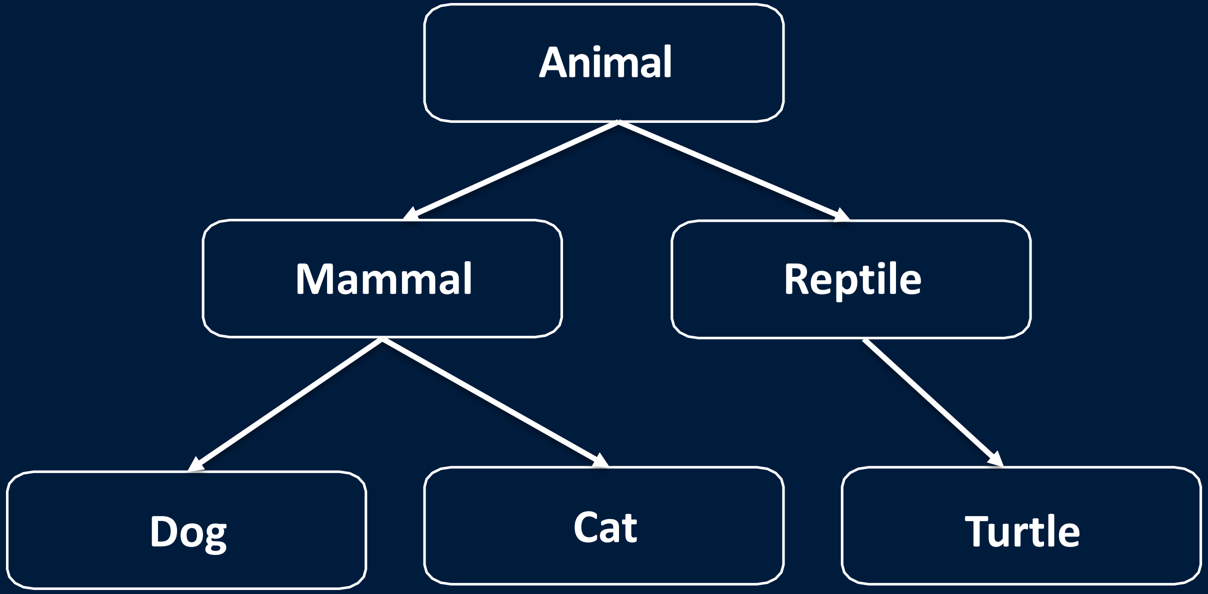
<Spot> <type> <Dog>.
<Oscar> <playsWith> <Spot>.
<Oscar> <type> <Cat>.



Class Taxonomies

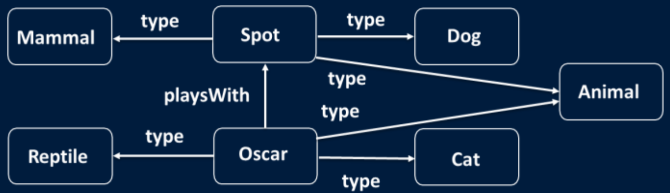
- Class taxonomies define a **<u>hierarchy</u>** between classes in a knowledge graph

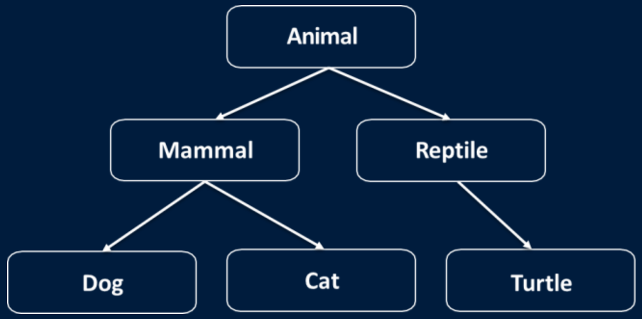
```
<owl:Class rdf:ID="Dog">
  <rdfs:subClassOf rdf:resource="#Mammal" />
</owl:Class>
<owl:Class rdf:ID="Cat">
  <rdfs:subClassOf rdf:resource="#Mammal" />
</owl:Class>
<owl:Class rdf:ID="Turtle">
  <rdfs:subClassOf rdf:resource="#Reptile" />
</owl:Class>
<owl:Class rdf:ID="Mammal">
  <rdfs:subClassOf rdf:resource="#Animal" />
</owl:Class>
<owl:Class rdf:ID="Reptile">
  <rdfs:subClassOf rdf:resource="#Animal" />
</owl:Class>
```



Problem

- How can we automatically induce a **class taxonomy** from an otherwise **flat knowledge graph**?



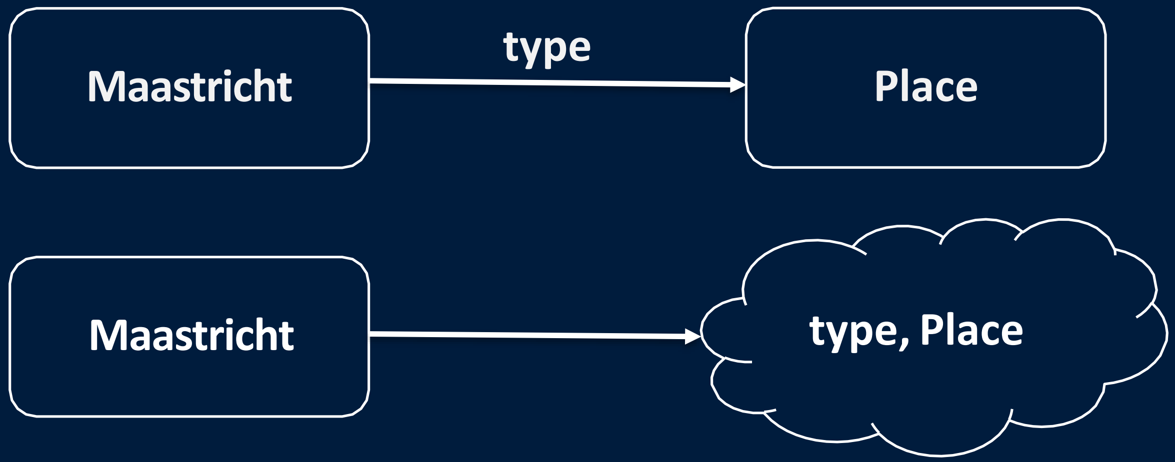


Problem

- How can we automatically induce a **class taxonomy** from an otherwise **flat knowledge graph**?

Restructuring

- Leverage the predicate which relates an entity to its class and **<u>restructure</u>** a knowledge graph to entities and tags
- Tags are defined as **<u>predicate-object pairs</u>**



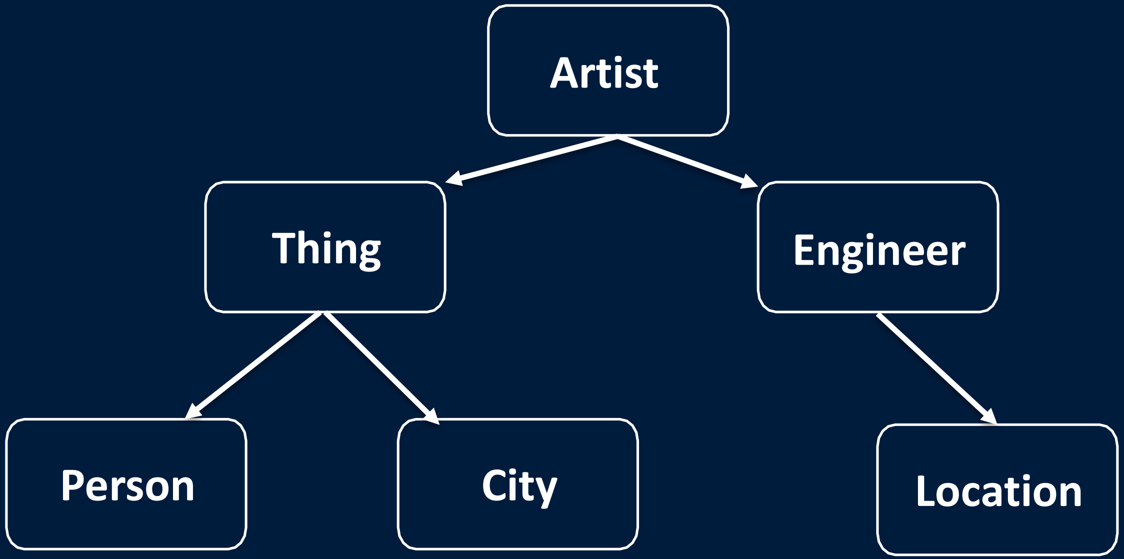
Proposed Model

- Restructuring the knowledge graph allows us to leverage ideas from topic models such as **Hierarchical Latent Dirichlet Allocation (hLDA)**

- Our proposed model is comprised of **three components**:
  - Subject paths (Nested Chinese Restaurant Process in hLDA)
  - Tag paths (Nested Chinese Restaurant Process in hLDA)
  - Tag levels (Stick-breaking Process in hLDA)

- The taxonomy is learned through **simulated annealing**
  - Local search through a space by generating **candidate solutions** and probabilistically accepting them

Tags create a hierarchy defined by their **paths** and **level allocations**

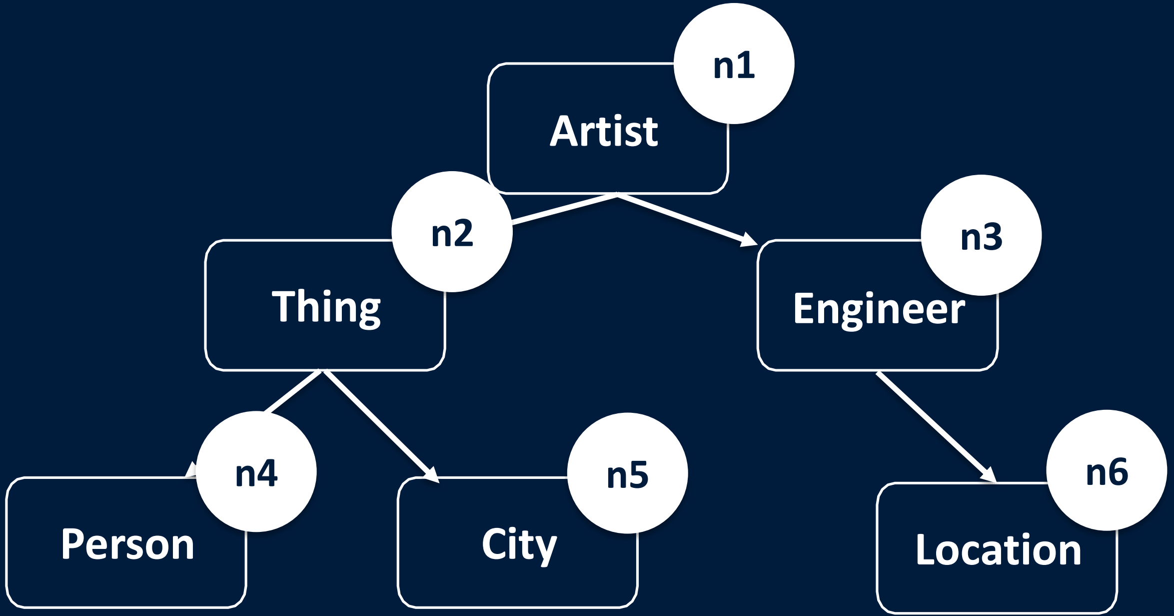


Tags create a hierarchy defined by their **paths** and **level allocations**

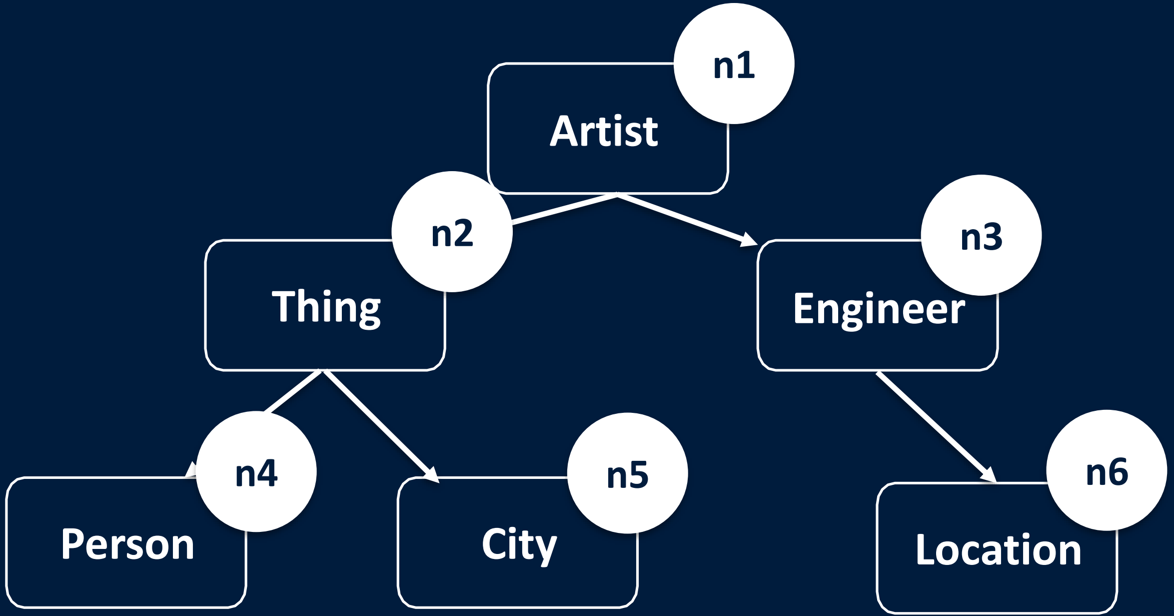
For example, the tag **Engineer** has:

Path: [n1, n3, n6]

Level: 2



Subjects are allocated to paths



London
Pablo Picasso
Salvador Dali
…

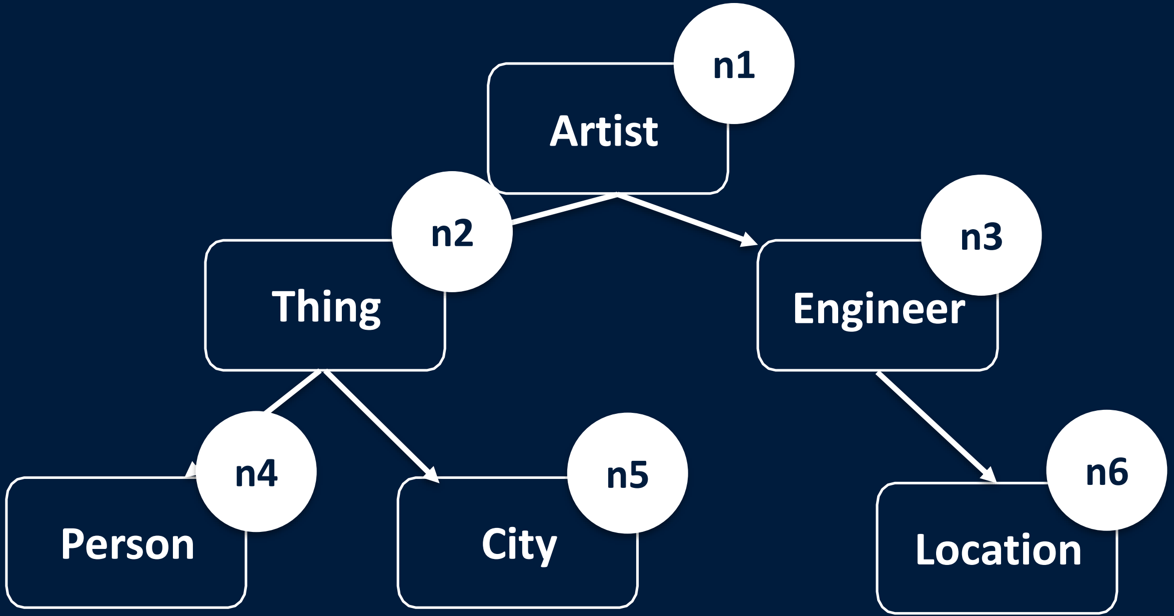
Maastricht
Vincent van Gogh
Nikola Tesla
…

Amsterdam
Paris
Thomas Edison
…

Subjects are allocated to paths

For example, the subject **Nikola Tesla** has:

Path: [n1, n2, n5]



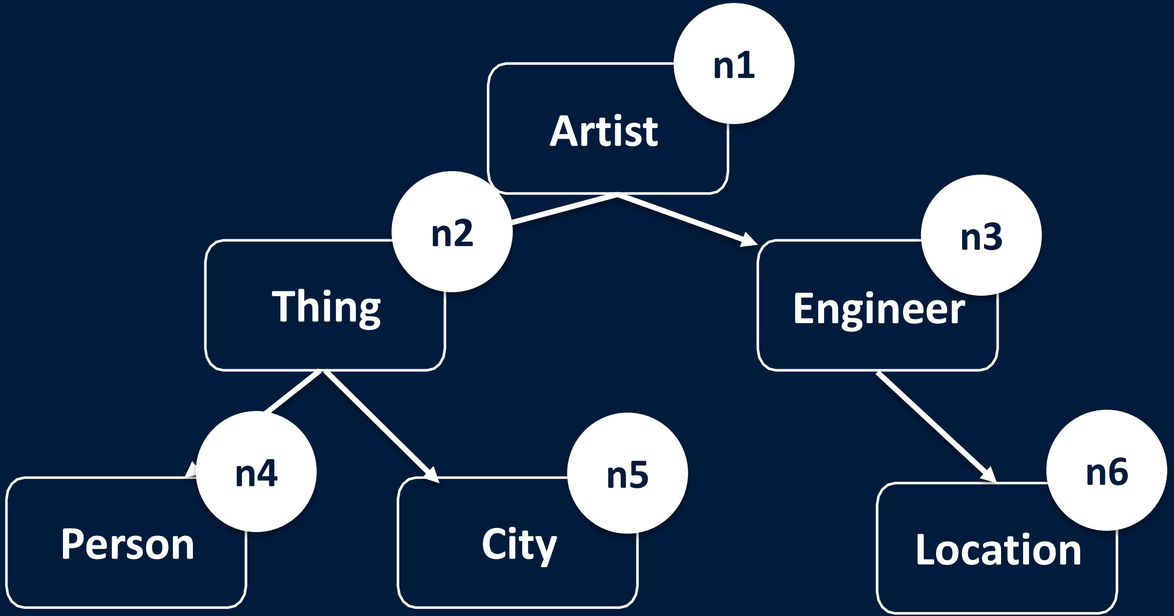
London
Pablo Picasso
Salvador Dali
…

Maastricht
Vincent van Gogh
Nikola Tesla
…

Amsterdam
Paris
Thomas Edison
…

To generate a new **candidate tree** we must:

1. Update subject paths
2. Update tag paths
3. Update tag levels

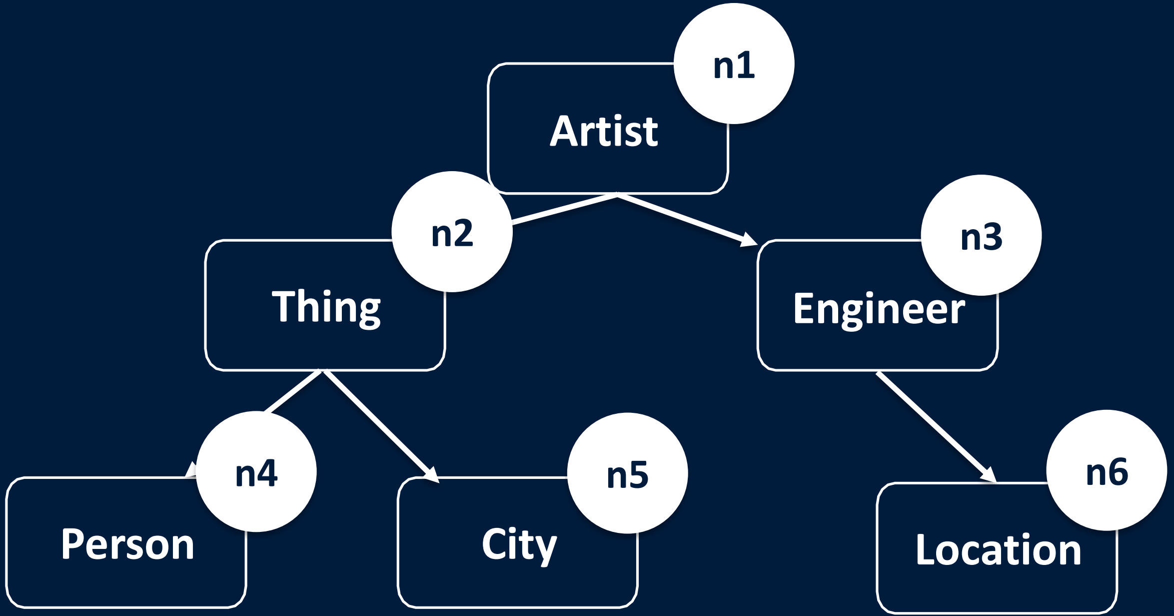


London
Pablo Picasso
Salvador Dali
…

Maastricht
Vincent van Gogh
Nikola Tesla
…

Amsterdam
Paris
Thomas Edison
…

Updating subject paths



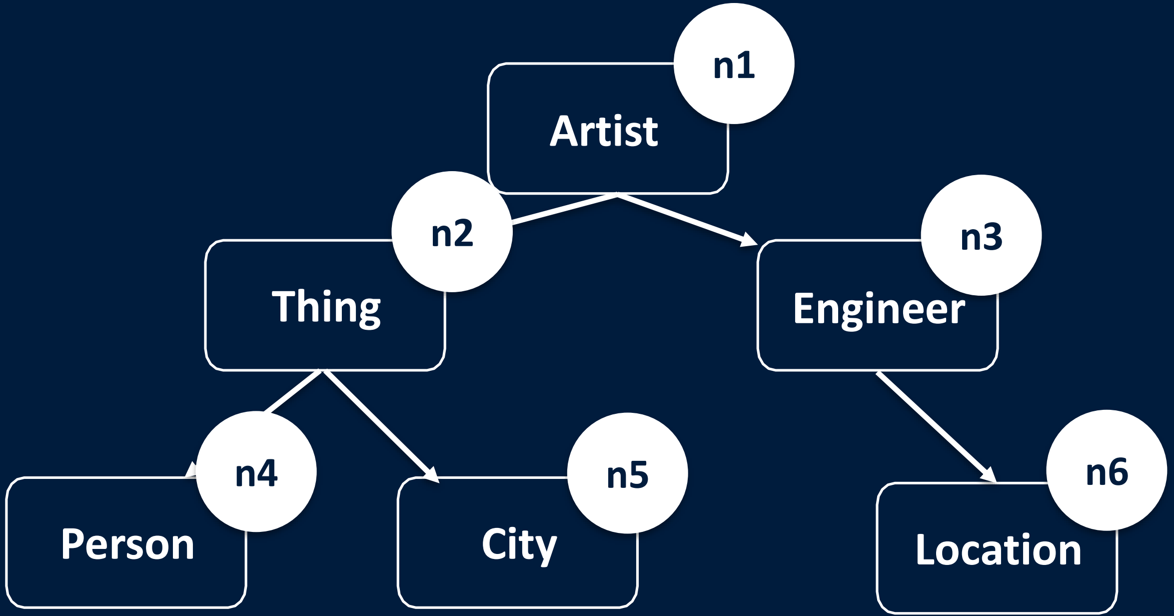
London
Pablo Picasso
Salvador Dali
…

Maastricht
Vincent van Gogh
Nikola Tesla
…

Amsterdam
Paris
Thomas Edison
…

Updating subject paths

Choose a subject at random. Let’s say **Nikola Tesla**



London
Pablo Picasso
Salvador Dali
…

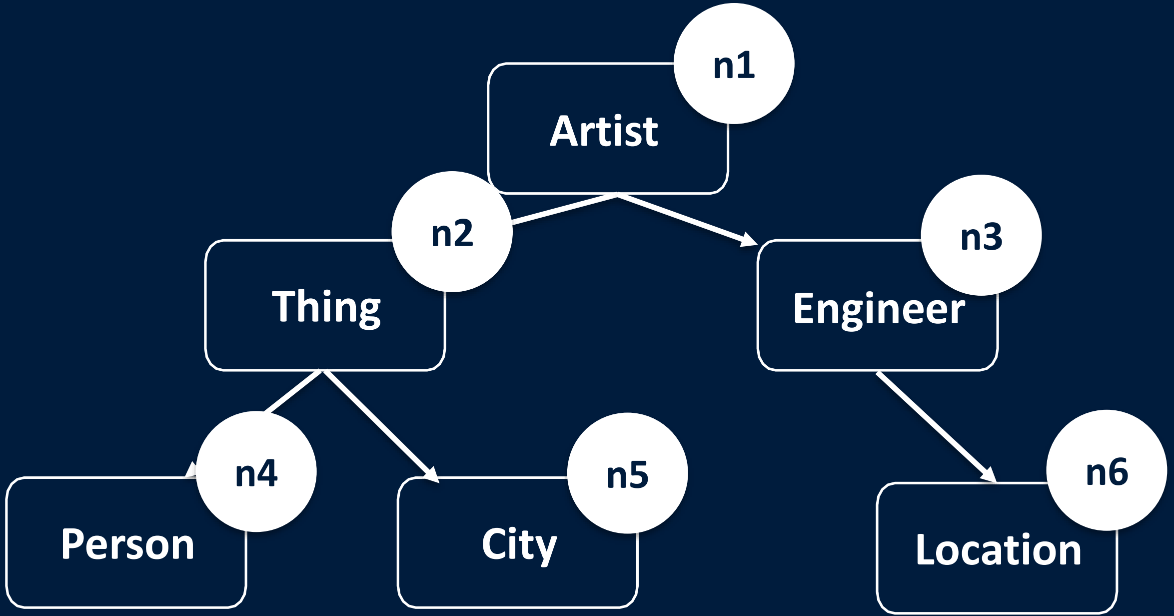
Maastricht
Vincent van Gogh
Nikola Tesla
…

Amsterdam
Paris
Thomas Edison
…

Updating subject paths

Choose a subject at random. Let's say **<u>Nikola Tesla</u>**

Remove **<u>Nikola Tesla</u>** from the tree



London	Maastricht	Amsterdam
Pablo Picasso	Vincent van Gogh	Paris
Salvador Dali		Thomas Edison
…	…	…

$$\hat{p}_i^S \sim \text{Multinomial}(\pi_k^S)$$

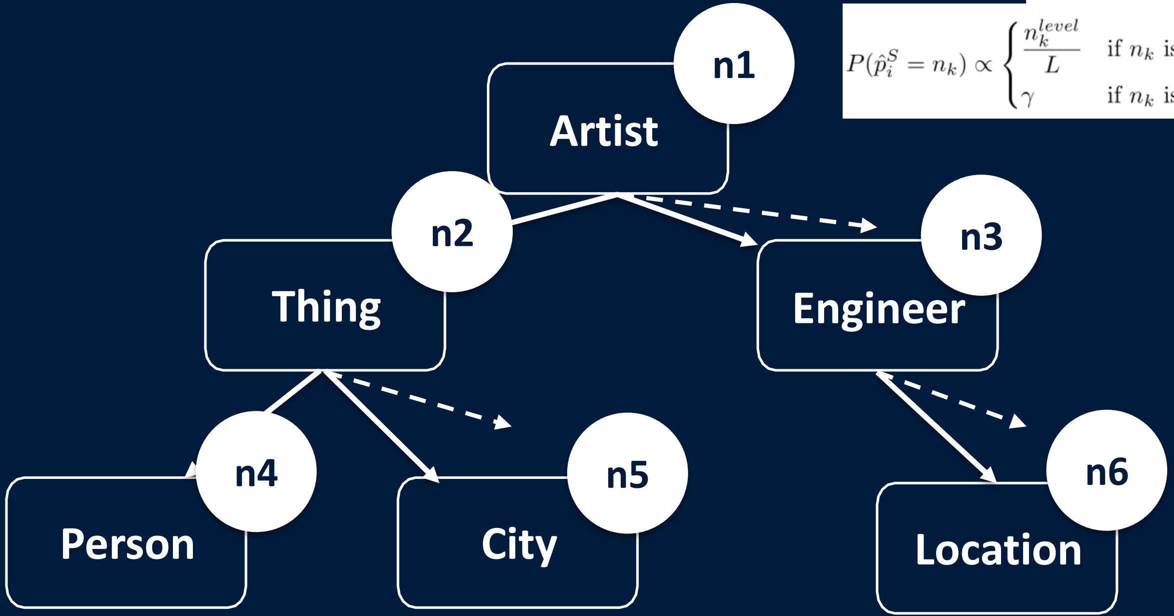
$$P(\hat{p}_i^S = n_k) \propto \begin{cases} \frac{n_k^{level}}{L} & \text{if } n_k \text{ is an internal node} \\ \gamma & \text{if } n_k \text{ is a leaf node} \end{cases}$$

Updating subject paths

Choose a subject at random. Let's say **Nikola Tesla**

Remove **Nikola Tesla** from the tree

Sample new path for **Nikola Tesla** using the update equations



London
Pablo Picasso
Salvador Dali
…

Maastricht
Vincent van Gogh
…

Amsterdam
Paris
Thomas Edison
…

$$\hat{p}_i^S \sim \text{Multinomial}(\pi_k^S)$$

$$P(\hat{p}_i^S = n_k) \propto \begin{cases} \frac{n_k^{level}}{L} & \text{if } n_k \text{ is an internal node} \\ \gamma & \text{if } n_k \text{ is a leaf node} \end{cases}$$

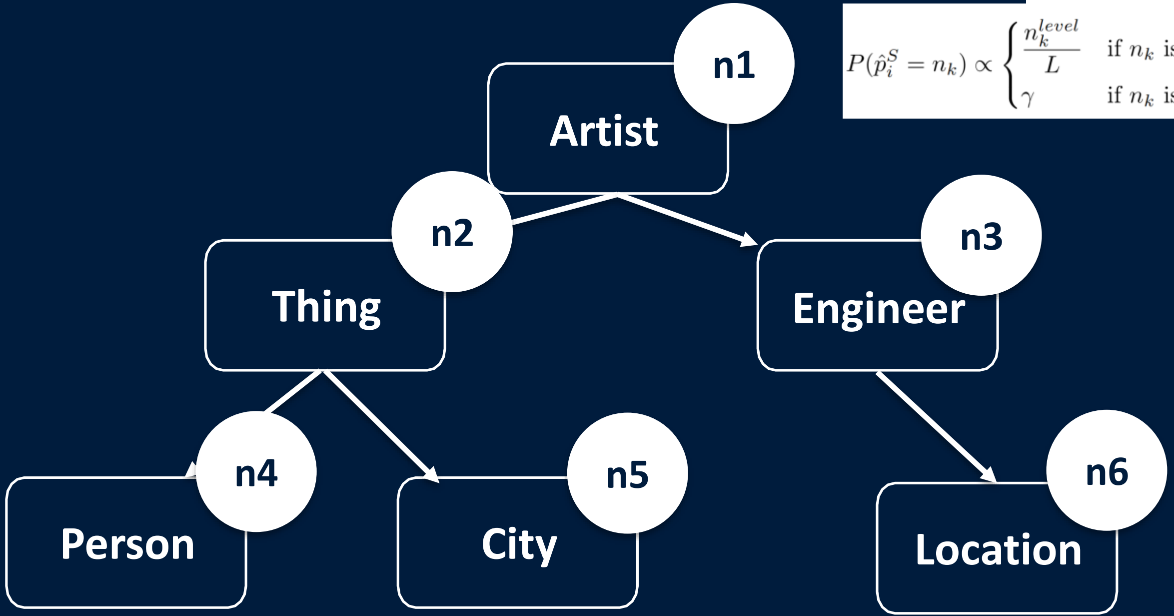
Updating subject paths

Choose a subject at random. Let's say **Nikola Tesla**

Remove **Nikola Tesla** from the tree

Sample new path for **Nikola Tesla** using the update equations

Let's say we sample path [n1, n2, n4]

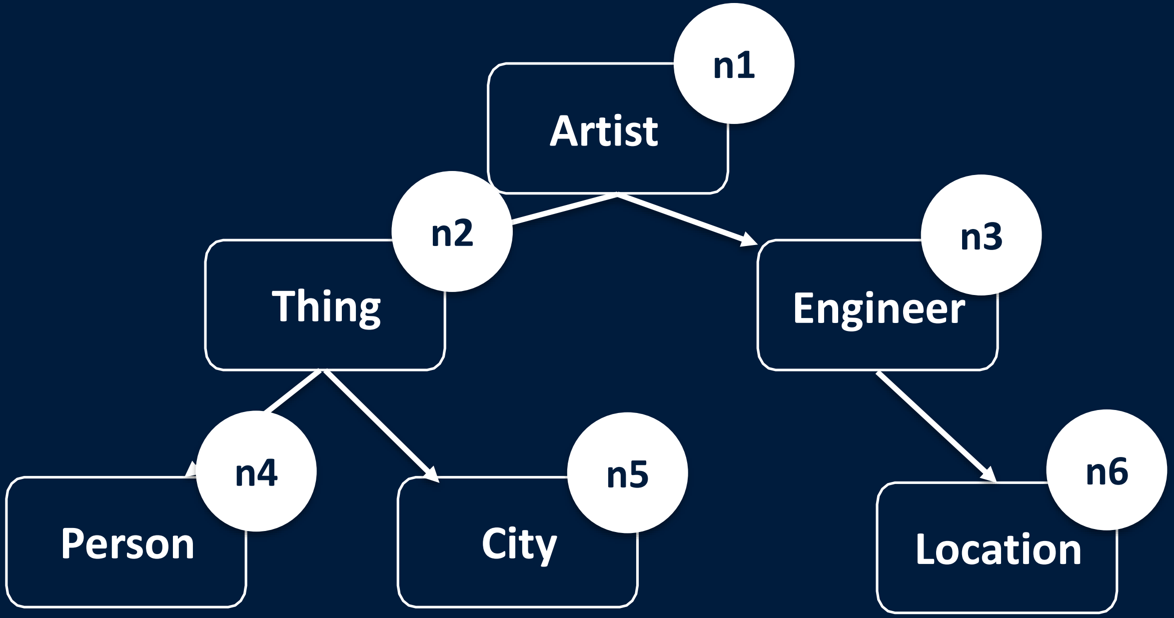


London
Pablo Picasso
Salvador Dali
Nikola Tesla
…

Maastricht
Vincent van Gogh
…

Amsterdam
Paris
Thomas Edison
…

Updating tag paths



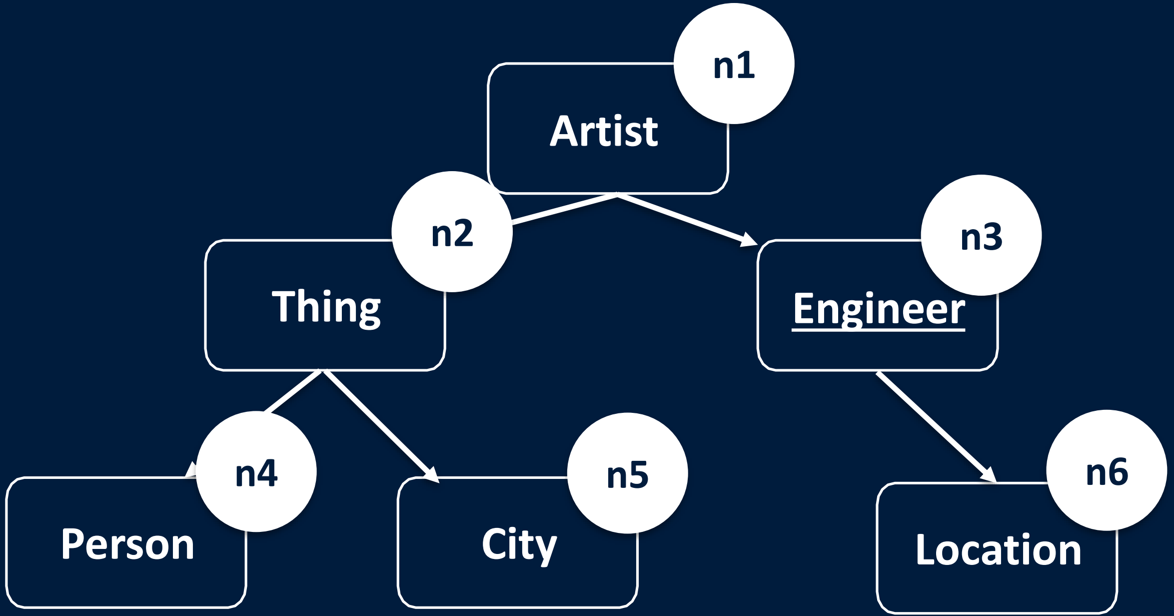
London
Pablo Picasso
Salvador Dali
Nikola Tesla
…

Maastricht
Vincent van Gogh
…

Amsterdam
Paris
Thomas Edison
…

Updating tag paths

As before choose a tag at random. Let's say **Engineer**



London
Pablo Picasso
Salvador Dali
Nikola Tesla
…

Maastricht
Vincent van Gogh
…

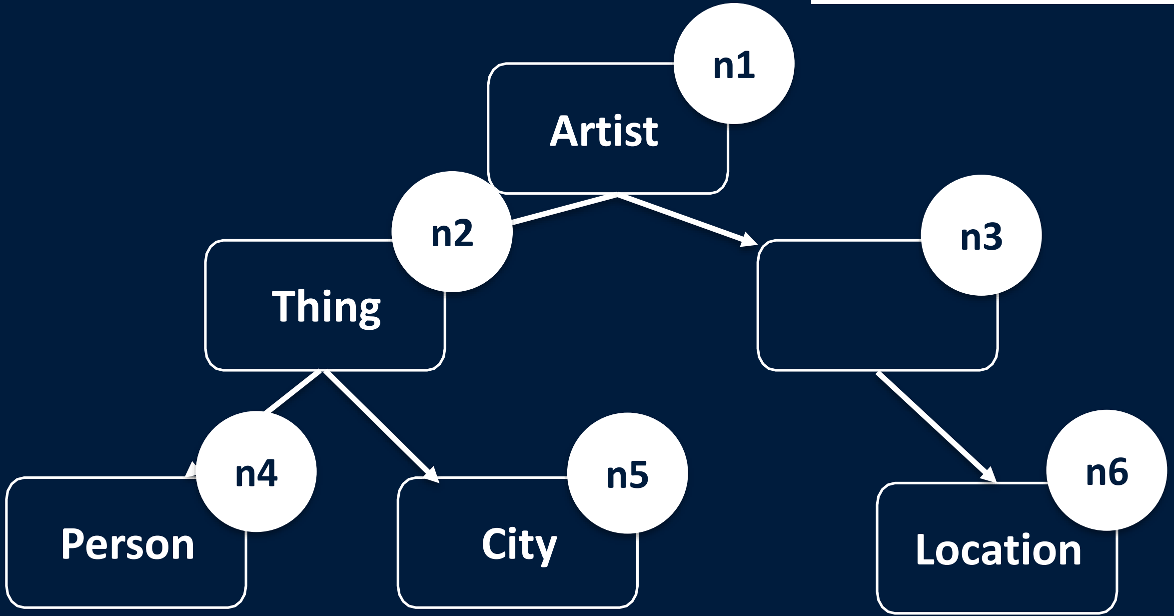
Amsterdam
Paris
Thomas Edison
…

$$\mathbf{p}_j^T \sim \mathrm{Uniform}(\{\mathbf{p}_i^S \in \mathbf{P}_{-j}^S : t_j \in \mathcal{T}_{s_i}\})$$

Updating tag paths

As before choose a tag at random. Let's say **Engineer**

Sample new path for **Engineer** using update equation



London
Pablo Picasso
Salvador Dali
Nikola Tesla
…

Maastricht
Vincent van Gogh
…

Amsterdam
Paris
Thomas Edison
…

$$\mathbf{p}_j^T \sim \text{Uniform}(\{\mathbf{p}_i^S \in \mathbf{P}_{-j}^S : t_j \in \mathcal{T}_{s_i}\})$$

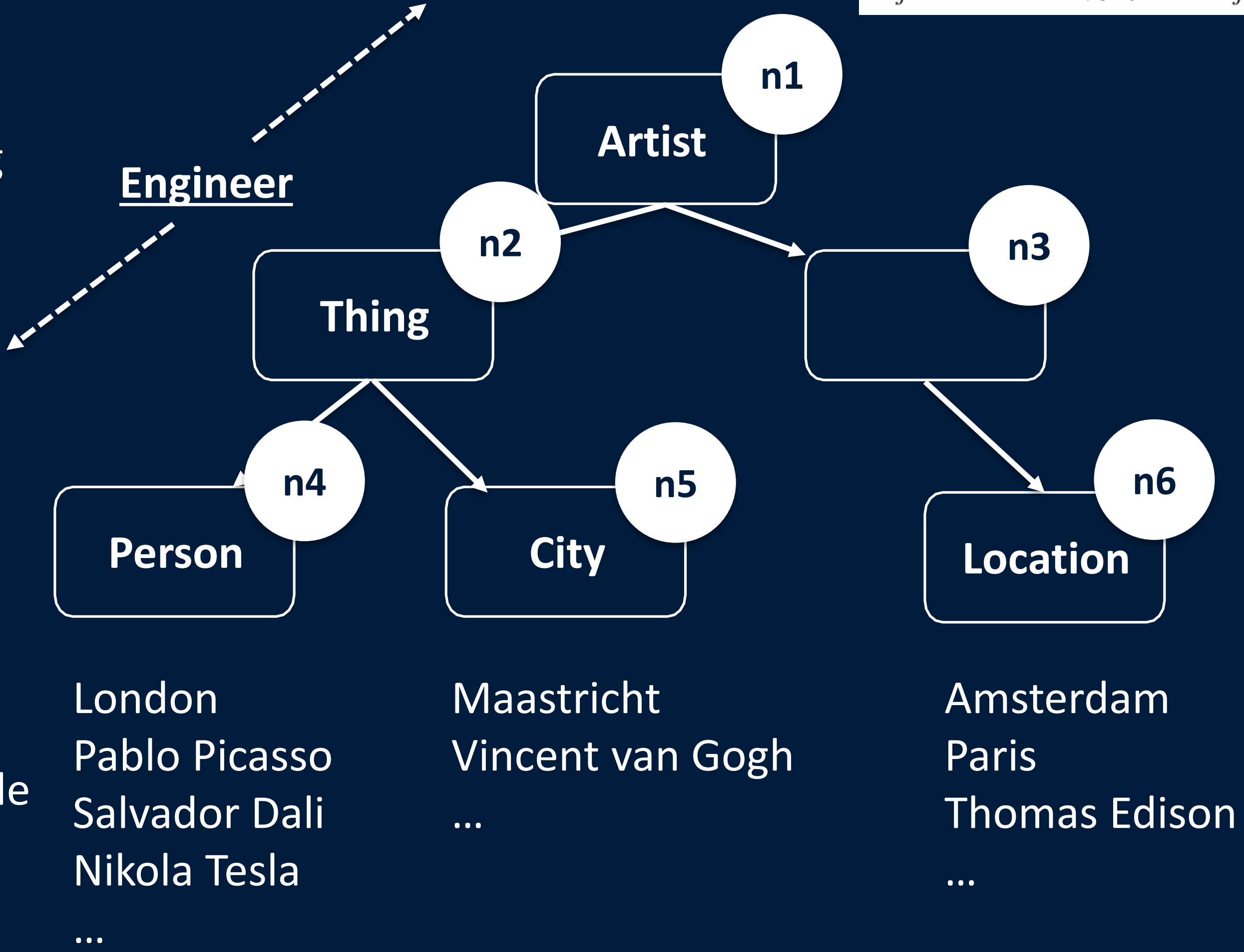
Updating tag paths

As before choose a tag at random. Let's say <u>**Engineer**</u>

Sample new path for <u>**Engineer**</u> using update equation

Let's say we sample path: [n1, n2, n4]

We now have to sample a tag level

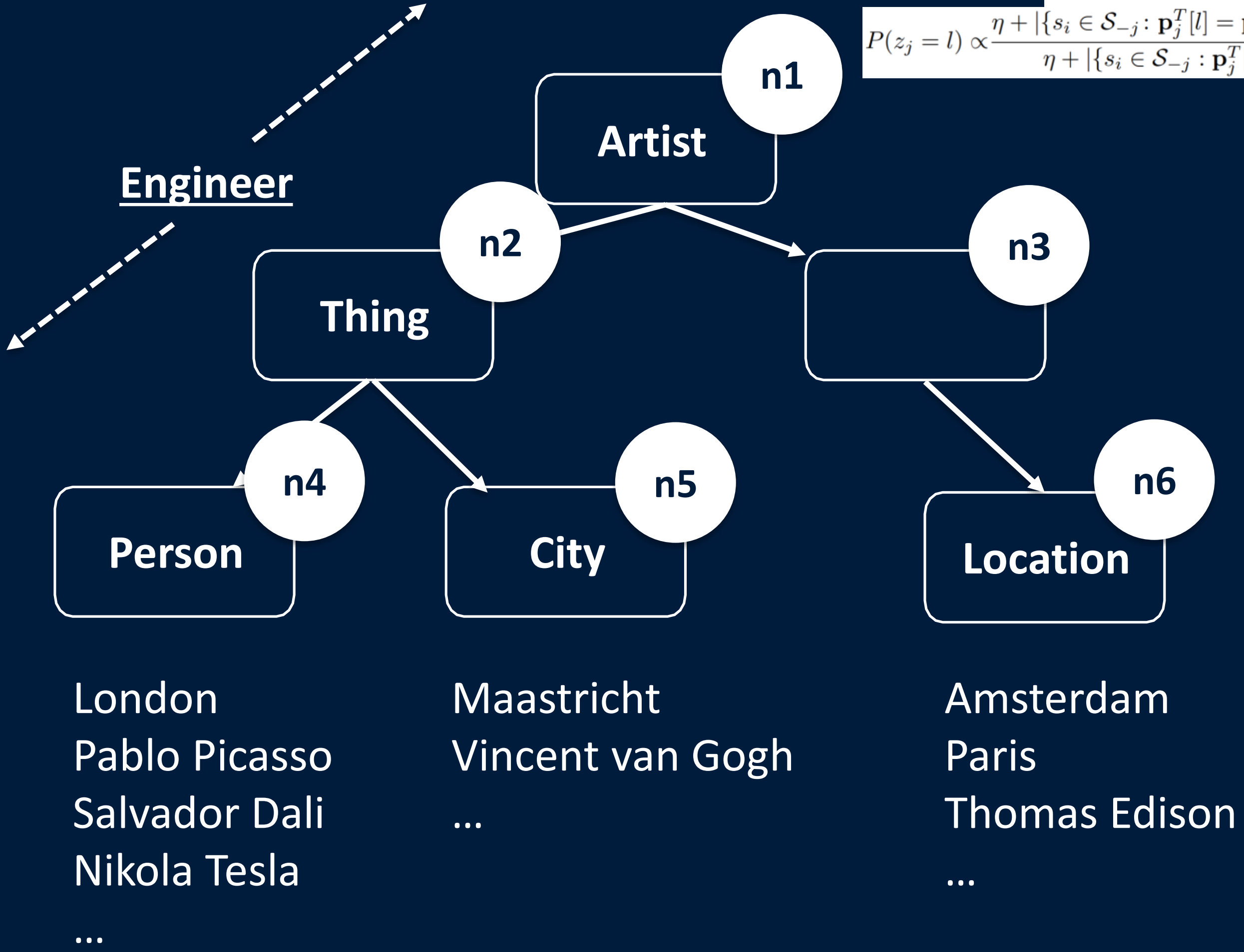


$$z_j \sim \mathrm{Multinomial}(\pi_k^T)$$

$$P(z_j = l) \propto \frac{\eta + |\{s_i \in \mathcal{S}_{-j} : \mathbf{p}_j^T[l] = \mathbf{p}_i^S[l] \wedge t_j \in \mathcal{T}_{s_i}\}|}{\eta + |\{s_i \in \mathcal{S}_{-j} : \mathbf{p}_j^T[l] = \mathbf{p}_i^S[l]\}|}$$

Updating tag levels

Sample a level for **Engineer** using the update equations



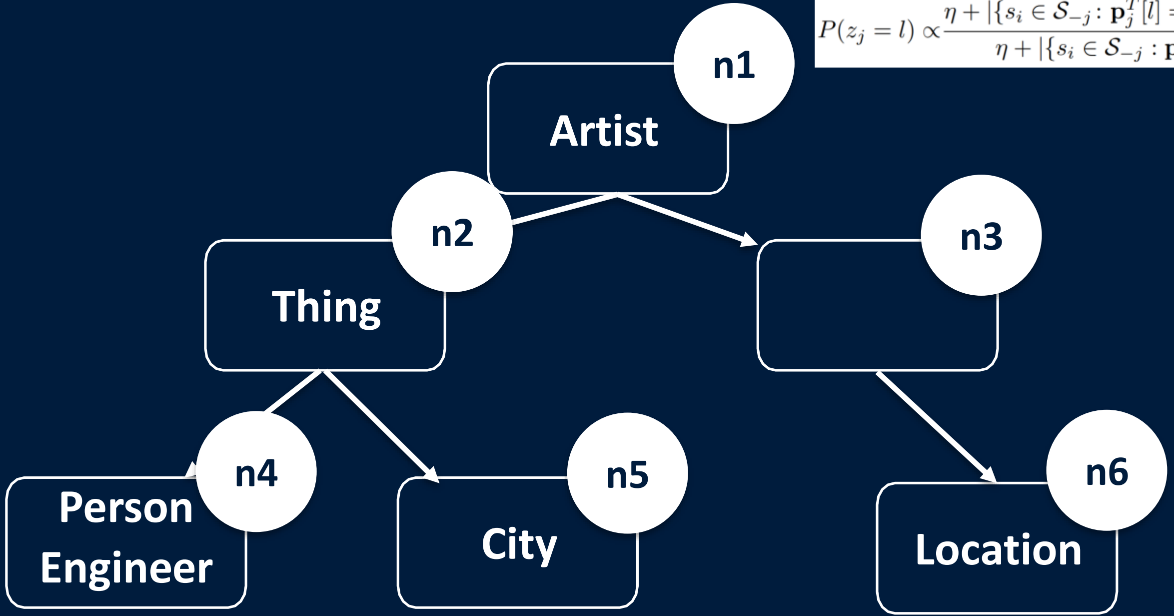
$$z_j \sim \mathrm{Multinomial}(\pi_k^T)$$

$$P(z_j = l) \propto \frac{\eta + |\{s_i \in \mathcal{S}_{-j} : \mathbf{p}_j^T[l] = \mathbf{p}_i^S[l] \wedge t_j \in \mathcal{T}_{s_i}\}|}{\eta + |\{s_i \in \mathcal{S}_{-j} : \mathbf{p}_j^T[l] = \mathbf{p}_i^S[l]\}|}$$

Updating tag levels

Sample a level for **Engineer** using the update equations

Let's say we sample level 3



London
Pablo Picasso
Salvador Dali
Nikola Tesla
…

Maastricht
Vincent van Gogh
…

Amsterdam
Paris
Thomas Edison
…

$$z_j \sim \text{Multinomial}(\pi_k^T)$$

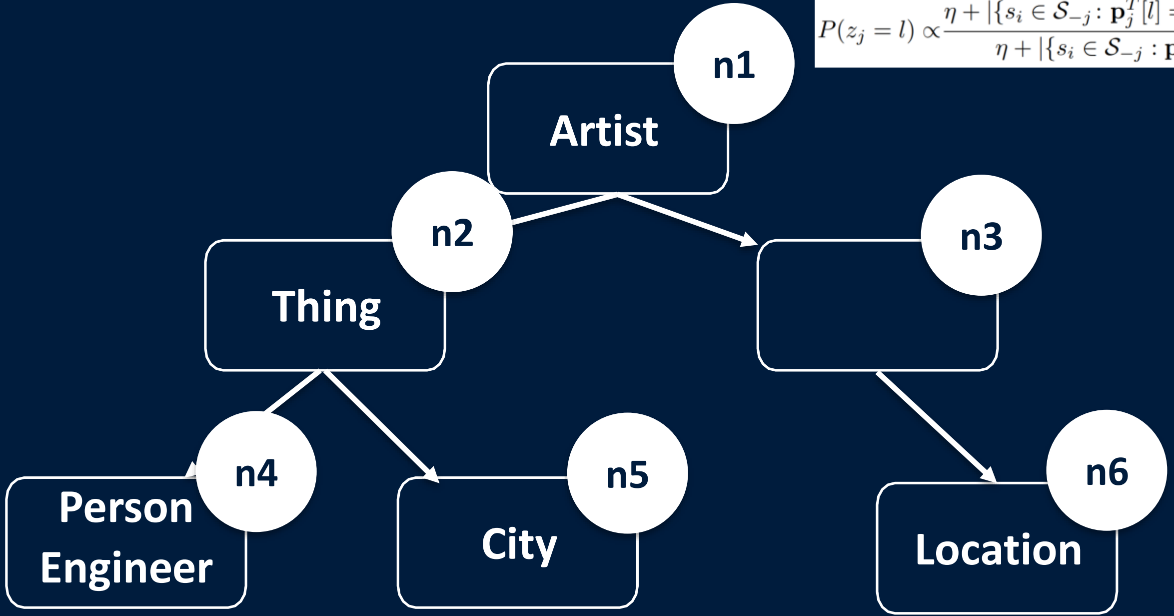
$$P(z_j = l) \propto \frac{\eta + |\{s_i \in \mathcal{S}_{-j} : \mathbf{p}_j^T[l] = \mathbf{p}_i^S[l] \wedge t_j \in \mathcal{T}_{s_i}\}|}{\eta + |\{s_i \in \mathcal{S}_{-j} : \mathbf{p}_j^T[l] = \mathbf{p}_i^S[l]\}|}$$

Updating tag levels

Sample a level for **Engineer** using the update equations

Let’s say we sample level 3

We have now generated a local neighbor, i.e. a **candiadate solution**



London
Pablo Picasso
Salvador Dali
Nikola Tesla
…

Maastricht
Vincent van Gogh
…

Amsterdam
Paris
Thomas Edison
…

Simulated Annealing

- Candidate solutions are evaluated according to an **objective function**

$$J = \sum_{s_i \in \mathcal{S}} \sum_{n_k \in \mathbf{p}_\mathbf{i}^\mathbf{S}} \sum_{t_j \in \mathcal{T}_{n_k}} \begin{cases} \frac{1}{|\mathcal{T}_{n_k}|} & \text{if } t_j \in \mathcal{T}_{s_i} \\ 0 & \text{otherwise} \end{cases}$$

$$- \sum_{s_i \in \mathcal{T}} \sum_{n_k \in \mathbf{p}_i^S} \begin{cases} |\mathcal{T}_{n_k}| + \alpha & \text{if } \mathcal{T}_{n_k} \cap \mathcal{T}_{s_i} = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

$$- \sum_{s_i \in \mathbf{T}} \sum_{n_l \in \mathbf{p}_\mathbf{i}^\mathbf{S}} \begin{cases} |\mathcal{T}_{n_k}|(L - n_k^{level}) + \alpha & \text{if } |\mathcal{T}_{n_k}| > 1 \\ 0 & \text{otherwise} \end{cases}$$

… and **accepted** according to an **annealing schedule**

$$\mathbb{P}(x = x') = \exp(-\frac{J_x - J_{x'}}{\theta_{iter}})$$

$$\theta_{iter} = \theta_0 * \frac{\theta_{final}}{\theta_0}^{\frac{iter}{iter_{max}}}$$

Evaluation Setup

- Run our algorithm on three **real-world datasets**:
  - Carnivora
  - DBpedia
  - WordNet

- Calculate **F1-scores** obtained using induced subsumption axioms and **gold standard** axioms for each dataset

- Compare F1-scores with **baselines** implemented from the literature

Results

- Results are **comparable** or **better than** state-of-the-art approaches

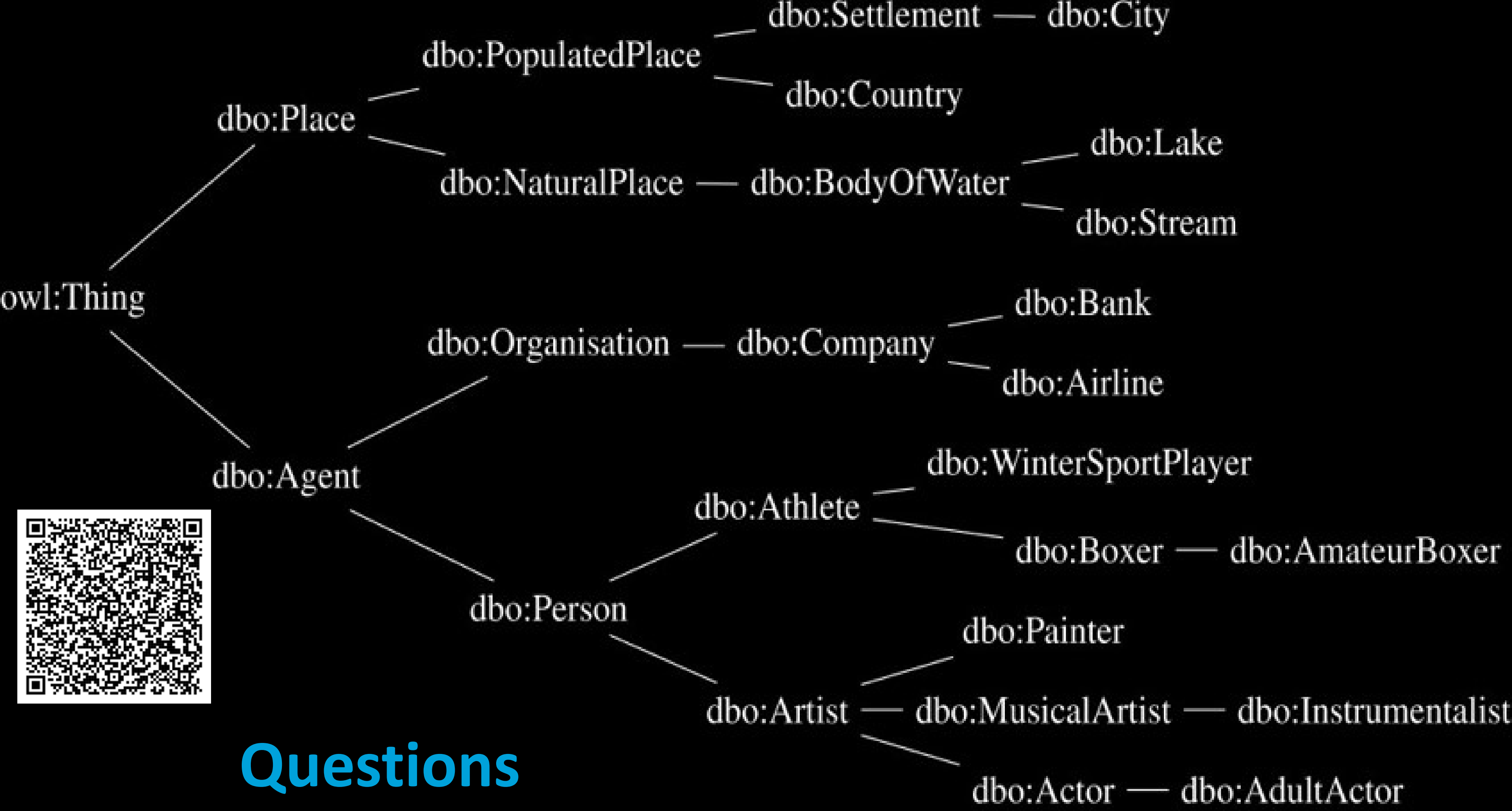
Model	Carnivora	Dbpedia	WordNet
Heymann and Garcia-Molina	0.9765	0.8673	0.5447
Schmitz	0.9831	0.8502	**0.6988**
Wang et al. (2012)	0.8571	0.6481	0.4462
Wang et al. (2018)	0.6908	0.5318	0.4286
Pietrasik and Reformat	0.9731	0.8788	0.6171
Our Model	**0.9866**	**0.8981**	0.5508

Conclusions and Future Work

- Path based models from topic modelling **can be used** for class taxonomy induction in knowledge graphs

- Biggest drawback of our model is that it is **slow**
  - Our model **does not scale** well to large datasets

- Future work will focus on **improving the scalability** of the model
  - Improve **sampling distributions** in update steps
  - Look into **alternative optimization schemes**


owl:Thing
dbo:Place
dbo:PopulatedPlace
dbo:Settlement
dbo:City
dbo:Country
dbo:NaturalPlace
dbo:BodyOfWater
dbo:Lake
dbo:Stream
dbo:Agent
dbo:Organisation
dbo:Company
dbo:Bank
dbo:Airline
dbo:Person
dbo:Athlete
dbo:WinterSportPlayer
dbo:Boxer
dbo:AmateurBoxer
dbo:Artist
dbo:Painter
dbo:MusicalArtist
dbo:Instrumentalist
dbo:Actor
dbo:AdultActor

Questions