Learning hierarchies from knowledge graphs

Marcin Pietrasik

UNIVERSITY
OF ALBERTA

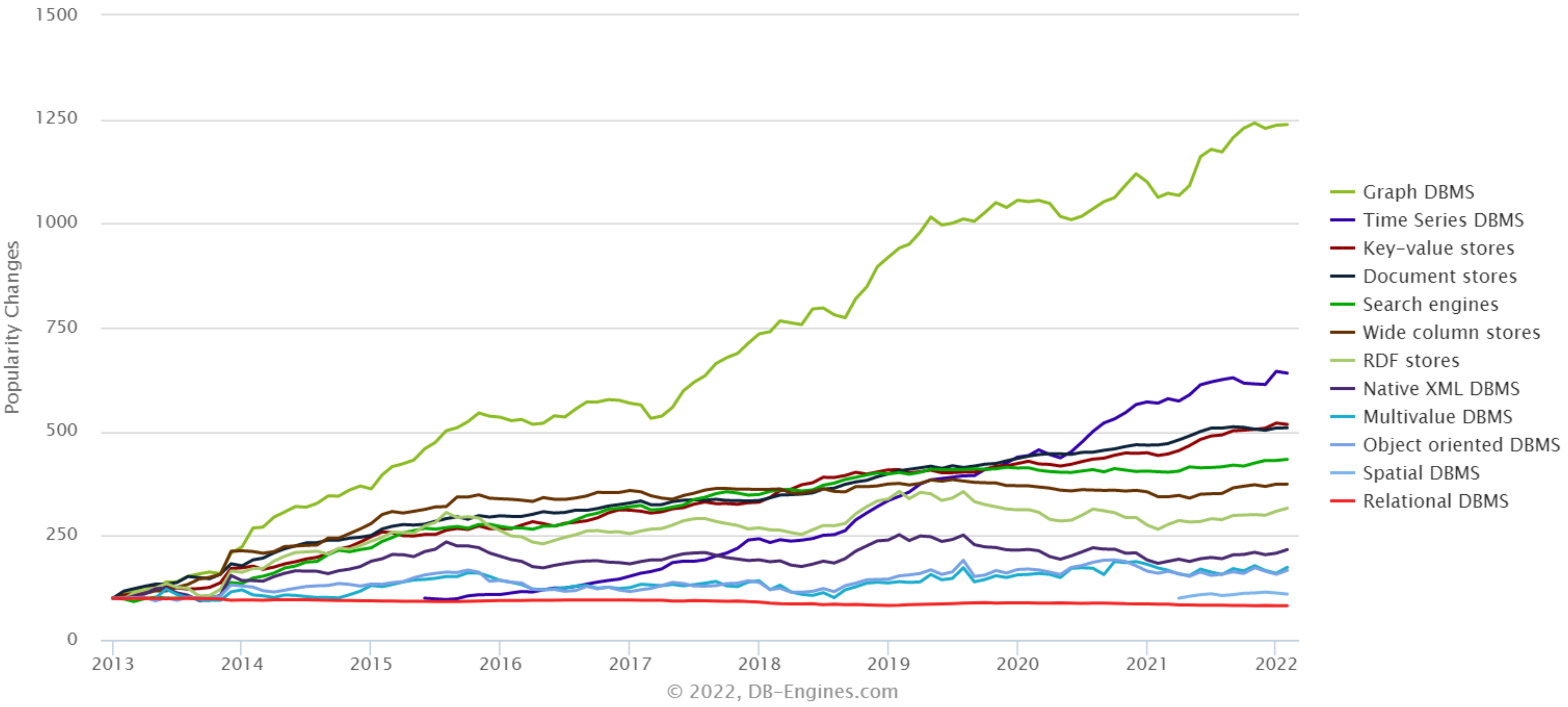
Outline

- Motivation
- Background
- Work to date: taxonomy induction from knowledge graphs
- Work to date: knowledge graph coarsening for embeddings
- Future work

Motivation

Motivation

Complete trend, starting with January 2013



UNIVERSITY OF ALBERTA

Motivation

- Knowledge graphs have garnered widespread attention in recent years in industry and academia alike
- Use cases include
  - Fraud detection
  - Social network modelling
  - Recommendation engines
  - Etc.

Knowledge graph hierarchies

- Why are hierarchies important to knowledge graphs?
  - Provide backbone and **structure** to the knowledge graph
  - Make knowledge graphs easier to **interpret** by viewing them at different levels of abstraction
  - Allow for inferring **relations** which are not otherwise explicit
  - **Aid** in solving common tasks related to knowledge graphs
  - Etc.

Expected contributions

- Develop a novel hierarchy induction technique based on frequencies and co-occurrences
- Adapt a hierarchical coarsening technique to improve knowledge graph embeddings
- Marry stochastic blockmodels with knowledge graphs to generate a hierarchy of concepts

Background

Knowledge graphs

- Knowledge graphs are a method of storing data as a graph structure
- Knowledge graphs are made up of **four** main **components:**
  - **Entities** (nodes, vertices, points, nouns, etc.)
  - **Predicates** (relations, edges, links, verbs, etc.)
  - **Literals**
  - **Blank nodes**

Triples

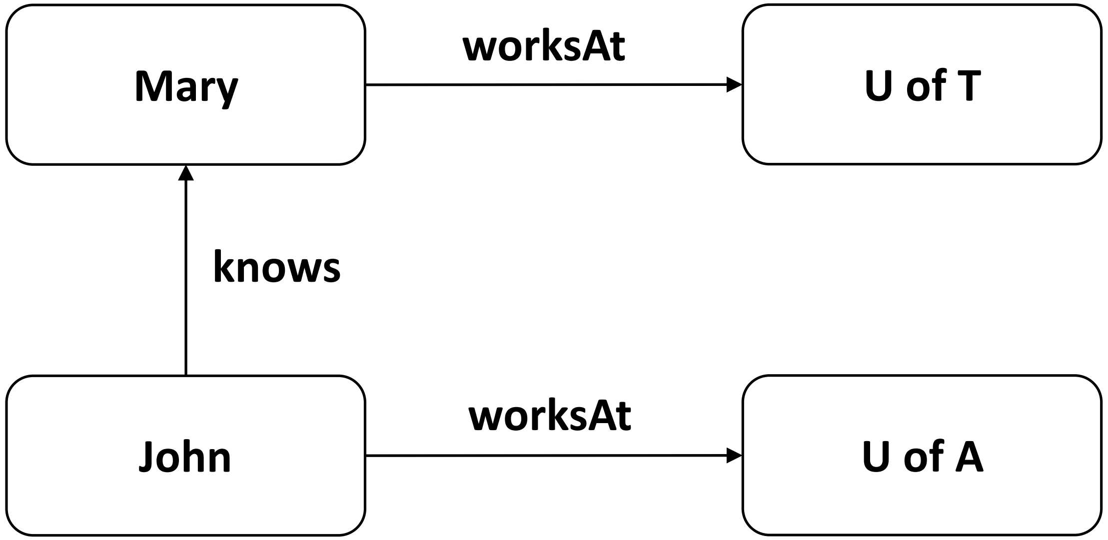
- Triples (facts) are how data is stored in a knowledge graph
  - Links a **subject** (head) to and **object** (tail) via a **predicate**

<John> <worksAt> <U of A>.

Triples

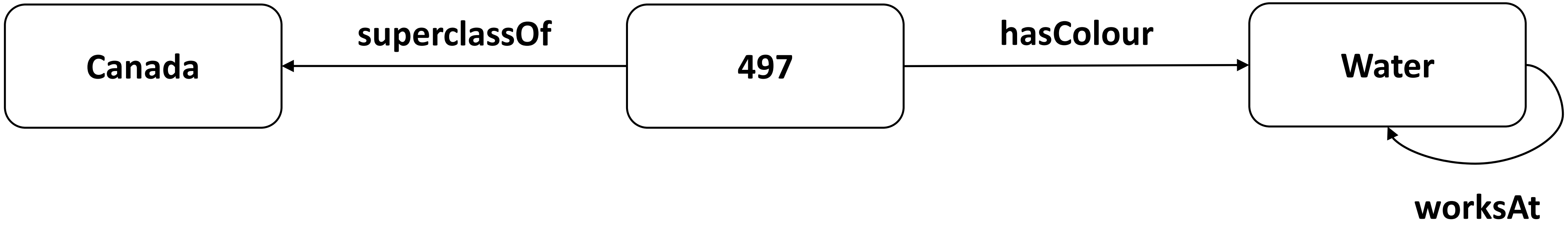
- Put triples together and you get a knowledge graph

<Mary> <worksAt> <U of T>.
<John> <knows> <Mary>.
<John> <worksAt> <U of A>.



Knowledge graphs store data only

- Does this make sense?



- We need **ontologies** to provide semantics!

Ontologies

- Ontologies provide **meaning** and **constraints** to knowledge graphs
    - Think of them as **rulebooks**
- Many use cases of ontologies:
    - Domain, range, subsumption, transitivity, symmetricity, cardinality, equivalence, set operations, enumeration, etc.

Ontology example

- Set the **domain** and **range** of predicates

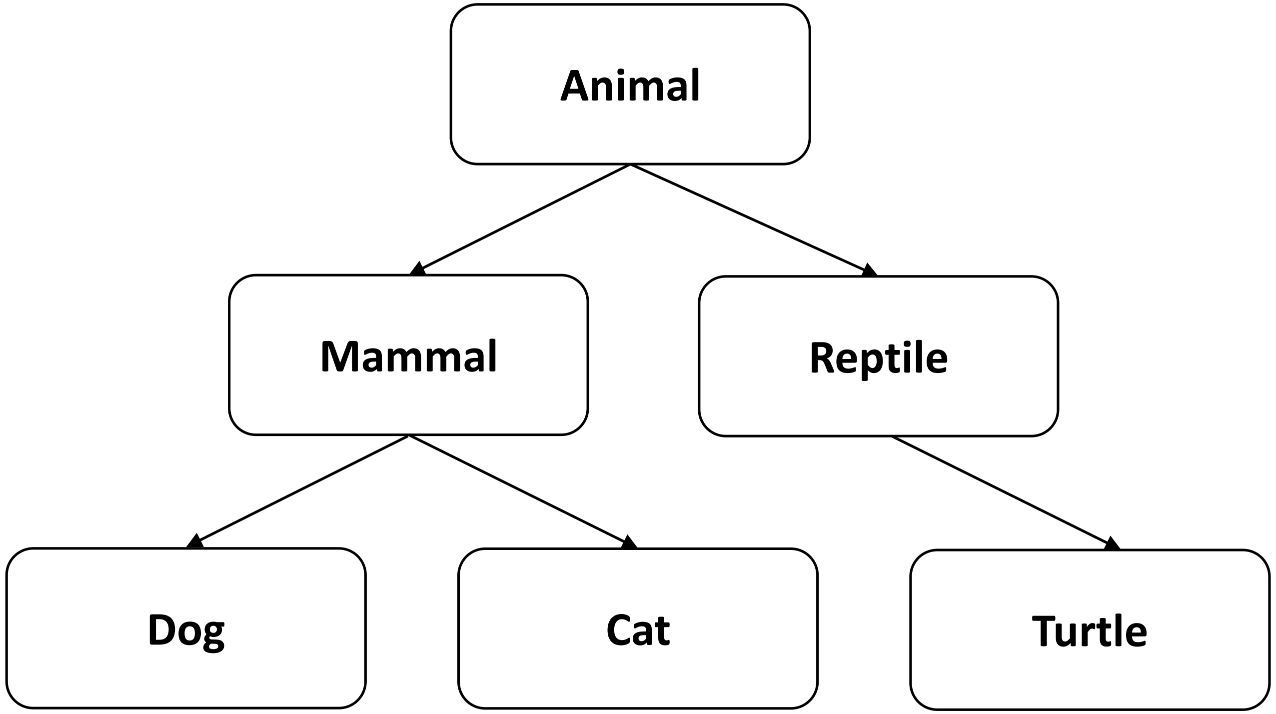
```
<owl:ObjectProperty rdf:ID="hasChild">
  <rdfs:domain rdf:resource="#Parent"/>
  <rdfs:range rdf:resource="#Child"/>
</owl:ObjectProperty>
```

- The **hasChild** predicate must relate a **parent** to a **child**

Ontology example

- Define a **hierarchy** between classes

```
<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>
```



Knowledge graph embeddings

- Embeddings represent a knowledge graph in a **continuous vector space**
- Reduce the **dimensionality** of the knowledge graph
- Mapping from discrete to continuous space opens the door to many **techniques** used in artificial intelligence

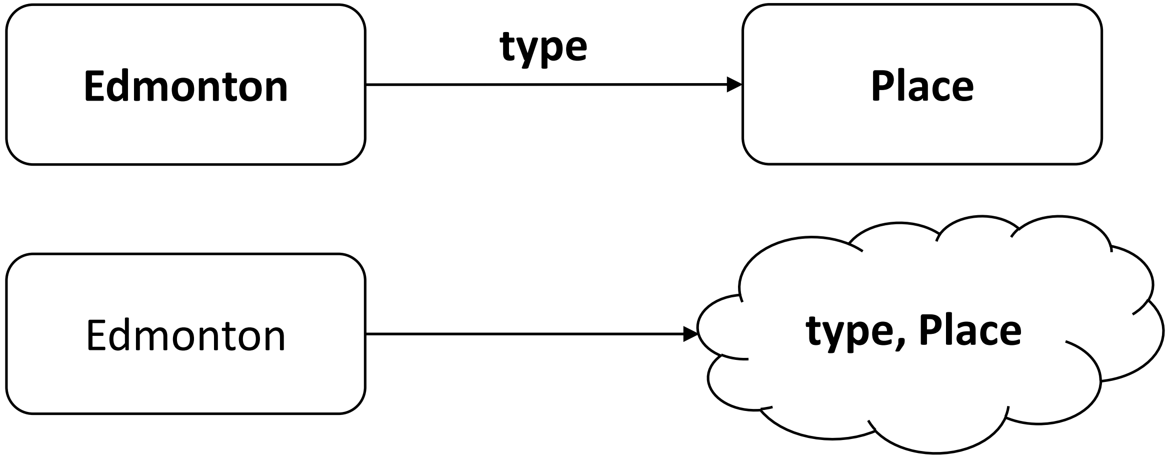
Work to date: taxonomy induction from knowledge graphs

Taxonomy induction

- A class taxonomy is a hierarchical structure which organizes a knowledge graph's classes through superclass-subclass relations
  - Generally a rooted tree or directed acyclic graph
- How to induce a class taxonomy from a flat knowledge graph?

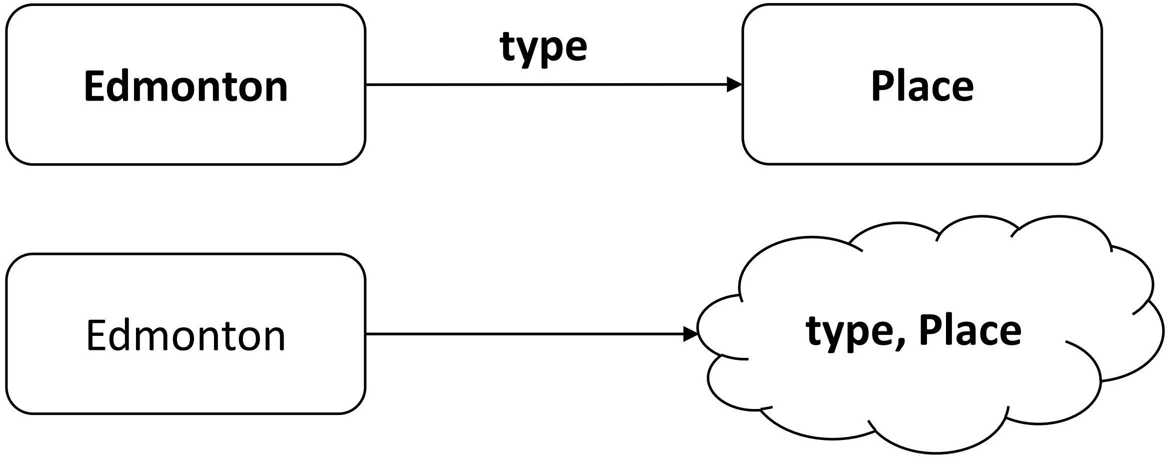
Proposed solution

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



Proposed solution

- When the knowledge graph is in a subject-tag structure, it opens the door to using **class frequencies** and **co-occurrences** to construct the taxonomy



Proposed solution

- Intuition
  - Classes which appear more often are more general and belong higher in taxonomy
  - Classes which describe the same subjects are closely related
- We need to calculate how often classes appear (**generality**) and how often two classes co-occur with one another (**similarity**)
- Having calculated the generality and similarity, the algorithm proceeds as follows:
  - Start the **most general** class as the root and greedily add classes to the taxonomy in decreasing generality
  - Classes are added as the child of the class they are **most similar** to

Results

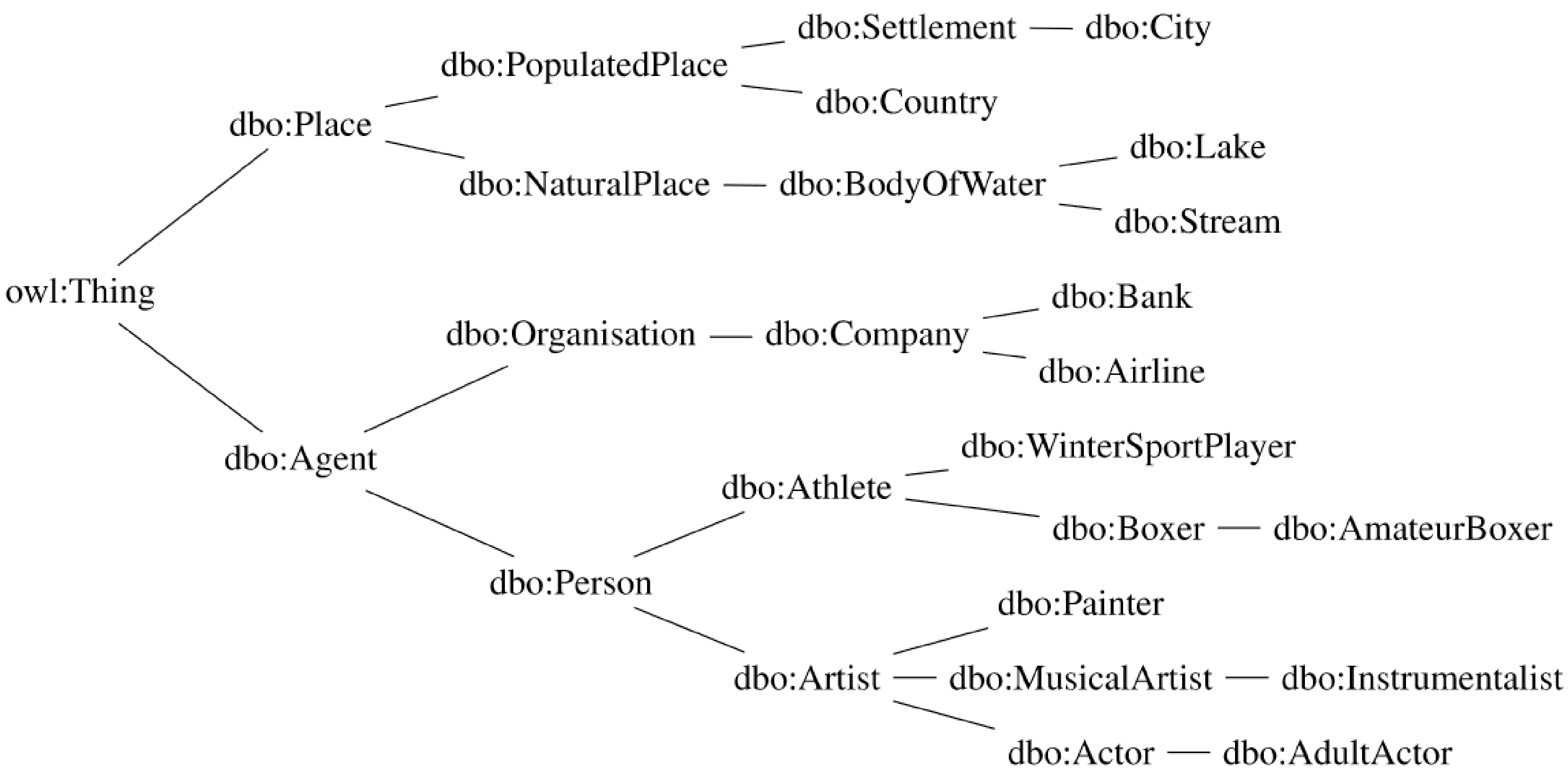


Fig. 2. Excerpts of the induced class taxonomies for the Life (top) and DBpedia (bottom) datasets. (Read left to right.)

UNIVERSITY OF ALBERTA

Evaluation

- The induced taxonomy can be compared to a gold standard taxonomy by calculating the **F_1 score** between each taxonomy's subsumption axioms

Model	Life	DBpedia	WordNet	IIMB
Heymann and Garcia		0.80	0.59	0.20
Schmitz	0.84	0.80	**0.79**	**0.52**
Paulheim and Fumkranz		0.14		
Ristoski et al.		0.52		
Zouaq and Martel		0.69		
Proposed Method	**0.86**	**0.88**	0.71	0.44

UNIVERSITY OF ALBERTA

Hierarchical clustering of subjects

- The induced taxonomy can serve as a backbone for a hierarchical clustering of knowledge graph subjects
- Intuition
  - Treat each tag in the taxonomy as a cluster and allocate subjects to the cluster it most belongs to
  - Belonging is calculated as the Jaccard coefficient between a subject's tags and the tags encountered in each cluster's path

Results

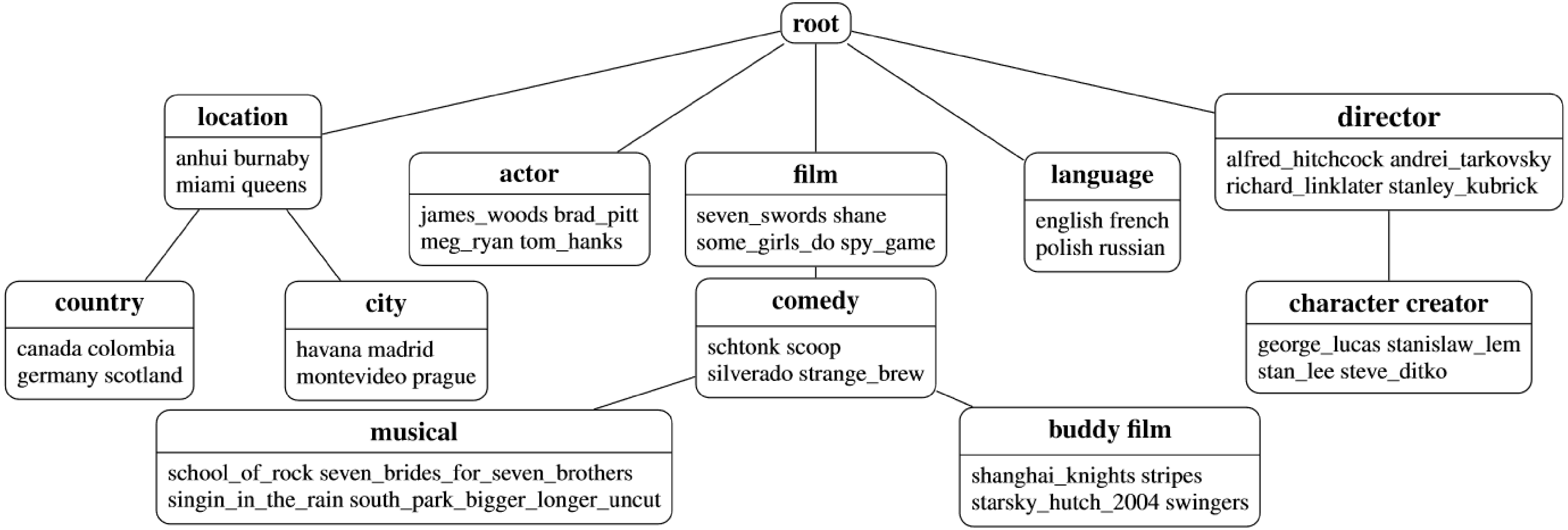


Fig. 3. Excerpt of the cluster hierarchy induced on the IIMB dataset. Node top indicates cluster's tag; bottom indicates cluster's constituent subjects.

UNIVERSITY OF ALBERTA

Work to date: knowledge graph coarsening for embeddings

UNIVERSITY OF ALBERTA

Graph coarsening

- Graph coarsening refers to the **merging** of entities in a graph that share similar structural properties
- It has been shown that coarsening as a preprocessing step can yield higher quality embeddings on **undirected** graphs
  - Can this also be the case for knowledge graphs?

UNIVERSITY OF ALBERTA

Proposed strategy

- The proposed strategy for using graph coarsening to improve knowledge graph embeddings is summarized in **three steps**:

    1. Probabilistic graph coarsening
    2. Coarse graph embedding
    3. Reverse mapping and fine tuning

UNIVERSITY OF ALBERTA

Probabilistic graph coarsening

- Probabilistic graph coarsening **merges entities** in the base graph to **entity clusters** in the coarse graph
- Because the pairwise comparison of all entities as candidates for merging is **expensive**, a probabilistic method is proposed

1. For each entity in the knowledge graph
    1. Sample *k* second order neighbours
    2. Sample *k* first order neighbours
2. For each entity in knowledge graph
    1. Merge second order neighbours if structurally similar
    2. Merge first order neighbours if structurally similar

Embedding and reverse mapping

- The coarse graph is embedded using a predetermined embedding method to obtain **coarse embeddings**
- Coarse embeddings are then **mapped** back down to the base graph
- To account for merged entities having the same embeddings, **fine tuning** is performed by running the embedding method on the base graph using coarse embeddings as **initializations**

Visualization

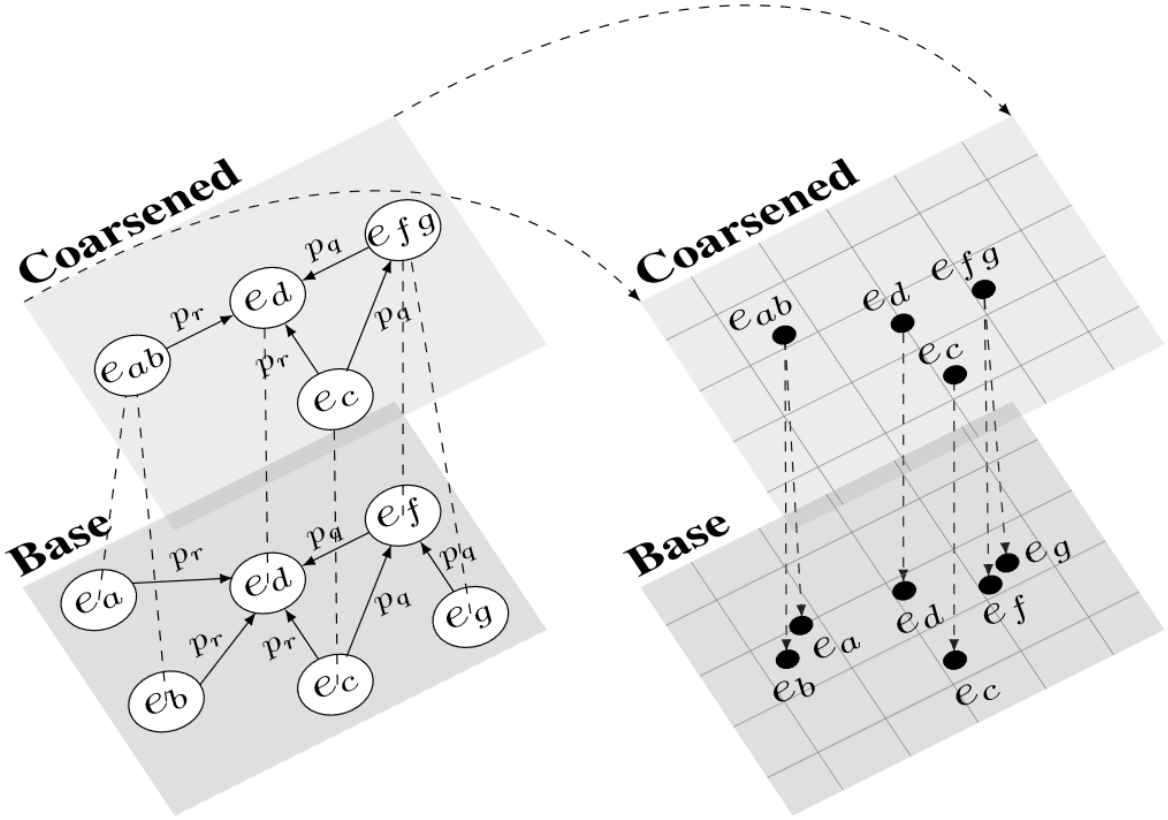


Figure 1: Toy example demonstrating our proposed embedding strategy. The logical flow is guided by dashed line arrows, starting in the bottom left corner and proceeding clockwise.

Evaluation

- The proposed strategy can be compared against embedding on the base graph
  - We examined **RDF2VEC**, **R-GCN**, and **TransE** embedding methods
- **Classification** is performed on learned embeddings to see how well they separate entities into different classes
  - **Accuracy** is used as the metric in our evaluation

Results

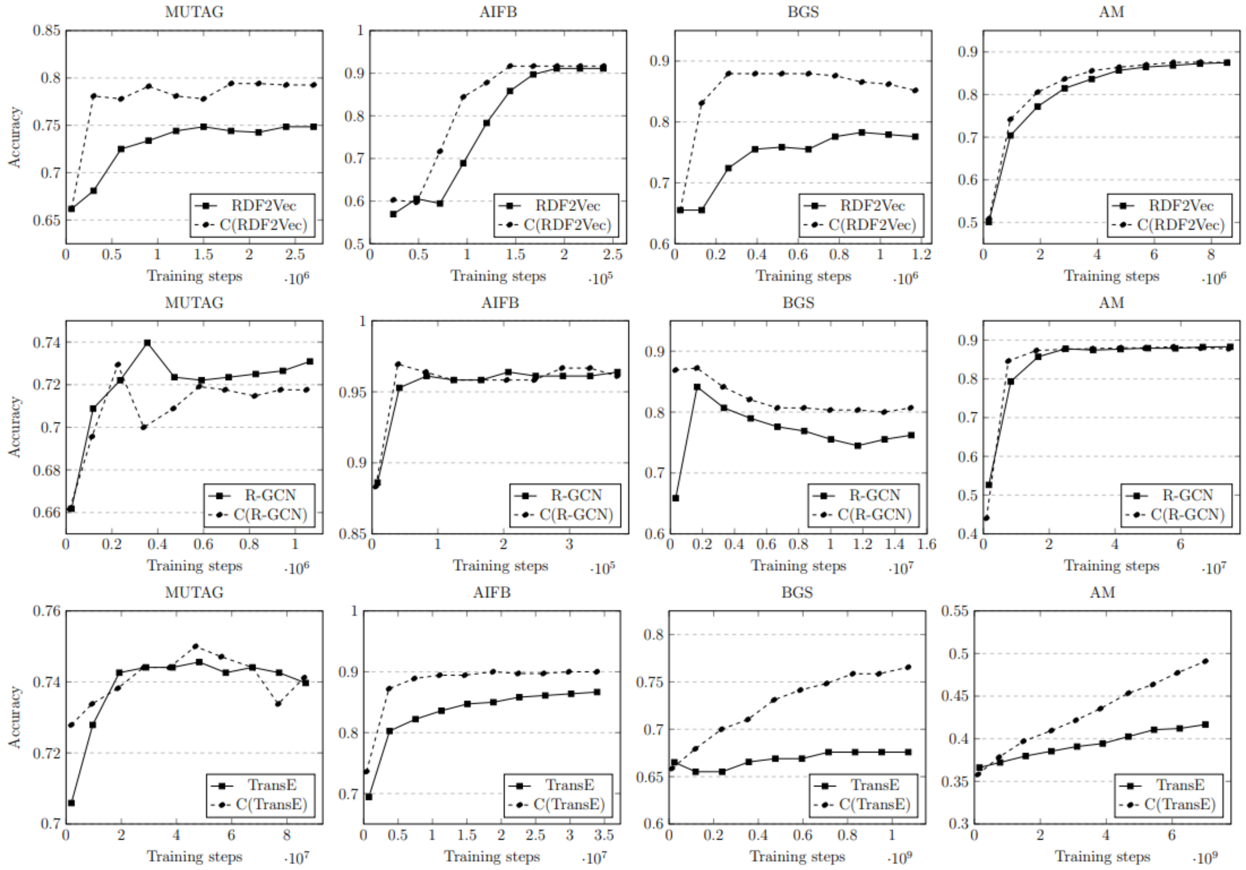


Figure 6: Pairwise comparison between baseline method and the proposed strategy demonstrating performance (accuracy) as a function of the number of training steps performed for each dataset.

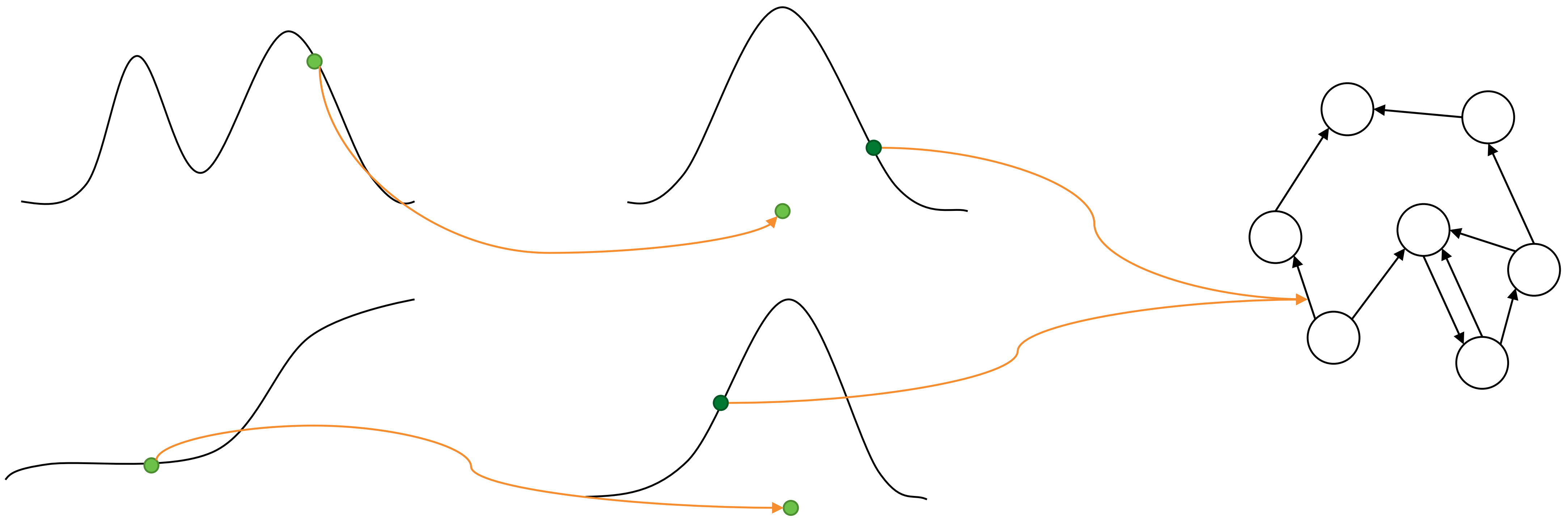
UNIVERSITY OF ALBERTA

Future work

Stochastic blockmodels

- Stochastic blockmodels are **generative models** for graphs
- Decompose a graph into **probability distributions**, or **blocks** of the model
- When **sampled** from, the blocks generate the graph
- The learning process is then to infer the parameters of these distributions

Stochastic blockmodels



Samples from blocks become means of subsequent blocks

Samples from blocks are used to predict the value of relations in network

UNIVERSITY OF ALBERTA

Gibbs sampling

- **Markov chain Monte Carlo** method for approximating a multivariate probability distribution
  - Used when it's easy (easier) to sample from **conditional distributions**
- Can be used to **infer** the parameters of a stochastic blockmodel
- General idea: iteratively sample from conditional distributions holding other parameters constant

Gibbs sampling

- Say you have two random variables, **A** and **B**, and you want to approximate the joint distribution, p(**A**,**B**) using Gibbs sampling:

1. Initialize **A** and **B** with some values
2. For *i* iterations
   1. Obtain new **A** from p(**A**|**B**)
   2. Obtain new **B** from p(**B**|**A**)

UNIVERSITY OF ALBERTA

Blockmodelling knowledge graphs

- Stochastic blockmodels can be modified to be used as a generative model for knowledge graphs
- When combined with a statistical prior over a tree structure, they can be used to generate hierarchies of communities from knowledge graphs
  - The **Nested Chinese Restaurant Process** is a stochastic process that can be used as a nonparametric prior over a tree structure

Blockmodelling knowledge graphs

- Stochastic blockmodels generate **communities** of entities and model the relations between these communities
- This has the effect of **abstracting** a knowledge graph into entities of similar concepts and modelling the relations between these concepts

Future work

- Continue work on coarsening as a tool for knowledge graph embeddings
- Develop a hierarchical stochastic blockmodel for knowledge graphs
- Investigate stochastic blockmodels as a method for abstracting knowledge graphs

Questions

UNIVERSITY OF ALBERTA