Lecture 1: Relational Data & Embedding Models

Relational Learning

Course Organisation

Course Organisation

Advanced Topics in Machine Learning: Research-oriented and somewhat less conventional course.

Course Organisation

Advanced Topics in Machine Learning: Research-oriented and somewhat less conventional course.

Themes of the lectures are:

Course Organisation

Advanced Topics in Machine Learning: Research-oriented and somewhat less conventional course.

Themes of the lectures are:

- **Relational Learning**: 8 lectures by *İsmail İlkan Ceylan* and 1 guest lecture.

Course Organisation

Advanced Topics in Machine Learning: Research-oriented and somewhat less conventional course.

Themes of the lectures are:

- **Relational Learning**: 8 lectures by *İsmail İlkan Ceylan* and 1 guest lecture.
- **Bayesian Machine Learning**: 8 lectures by *Atılım Güneș Baydın* and 1 guest lecture.

Course Organisation

Advanced Topics in Machine Learning: Research-oriented and somewhat less conventional course.

Themes of the lectures are:

- **Relational Learning**: 8 lectures by *İsmail İlkan Ceylan* and 1 guest lecture.
- **Bayesian Machine Learning**: 8 lectures by *Atılım Güneş Baydın* and 1 guest lecture.

For both themes, there will be a live-streamed guest lecture towards the end of the term. Please follow the announcements on the course website.

Course Organisation

Advanced Topics in Machine Learning: Research-oriented and somewhat less conventional course.

Themes of the lectures are:

- **Relational Learning**: 8 lectures by *İsmail İlkan Ceylan* and 1 guest lecture.
- **Bayesian Machine Learning**: 8 lectures by *Atılım Güneş Baydın* and 1 guest lecture.

For both themes, there will be a live-streamed guest lecture towards the end of the term. Please follow the announcements on the course website.

Assessment: Through a reproducibility challenge, as detailed in the assessment form available from the course webpage. More details will follow in due course.

Course Organisation

Advanced Topics in Machine Learning: Research-oriented and somewhat less conventional course.

Themes of the lectures are:

- **Relational Learning**: 8 lectures by *İsmail İlkan Ceylan* and 1 guest lecture.
- **Bayesian Machine Learning**: 8 lectures by *Atılım Güneș Baydın* and 1 guest lecture.

For both themes, there will be a live-streamed guest lecture towards the end of the term. Please follow the announcements on the course website.

Assessment: Through a reproducibility challenge, as detailed in the assessment form available from the course webpage. More details will follow in due course.

Practicals: There are 6 practicals planned. These practicals will provide you the necessary technical skills for the projects. Two of these practicals are specifically dedicated for discussing the assessment papers and helping you to form groups, depending on your interests.

Course Structure: Relational Learning

Course Structure: Relational Learning

The lectures cover *some* selected (and mostly recent) topics in relational learning. There are various subfields of relational learning which are not covered in this course.

Course Structure: Relational Learning

The lectures cover *some* selected (and mostly recent) topics in relational learning. There are various subfields of relational learning which are not covered in this course.

The (pre-recorded) lectures for relational learning are organised as follows:

Course Structure: Relational Learning

The lectures cover *some* selected (and mostly recent) topics in relational learning. There are various subfields of relational learning which are not covered in this course.

The (pre-recorded) lectures for relational learning are organised as follows:

- **Knowledge graphs and embedding models** (2 lectures)

Course Structure: Relational Learning

The lectures cover *some* selected (and mostly recent) topics in relational learning. There are various subfields of relational learning which are not covered in this course.

The (pre-recorded) lectures for relational learning are organised as follows:

- **Knowledge graphs and embedding models** (2 lectures)
- **Graph neural networks** (6 lectures)

Course Structure: Relational Learning

The lectures cover *some* selected (and mostly recent) topics in relational learning. There are various subfields of relational learning which are not covered in this course.

The (pre-recorded) lectures for relational learning are organised as follows:

- **Knowledge graphs and embedding models** (2 lectures)
- **Graph neural networks** (6 lectures)

There will be dedicated office hours (from Week 4, onwards) for your questions. Please follow the announcements regarding this.

Course Structure: Relational Learning

The lectures cover *some* selected (and mostly recent) topics in relational learning. There are various subfields of relational learning which are not covered in this course.

The (pre-recorded) lectures for relational learning are organised as follows:

- **Knowledge graphs and embedding models** (2 lectures)
- **Graph neural networks** (6 lectures)

There will be dedicated office hours (from Week 4, onwards) for your questions. Please follow the announcements regarding this.

These topics are covered for the first time in the scope of this course, and so the material is new. Please email me if you spot any problems in the slides and I will revise them accordingly.

Overview of the Lecture

Overview of the Lecture

- Relational data

Overview of the Lecture

- Relational data
- Knowledge graphs

Overview of the Lecture

- Relational data
- Knowledge graphs
- Knowledge graph embedding models

Overview of the Lecture

- Relational data
- Knowledge graphs
- Knowledge graph embedding models
- Model expressiveness

Overview of the Lecture

- Relational data
- Knowledge graphs
- Knowledge graph embedding models
- Model expressiveness
- Model inductive capacity & inference patterns

Overview of the Lecture

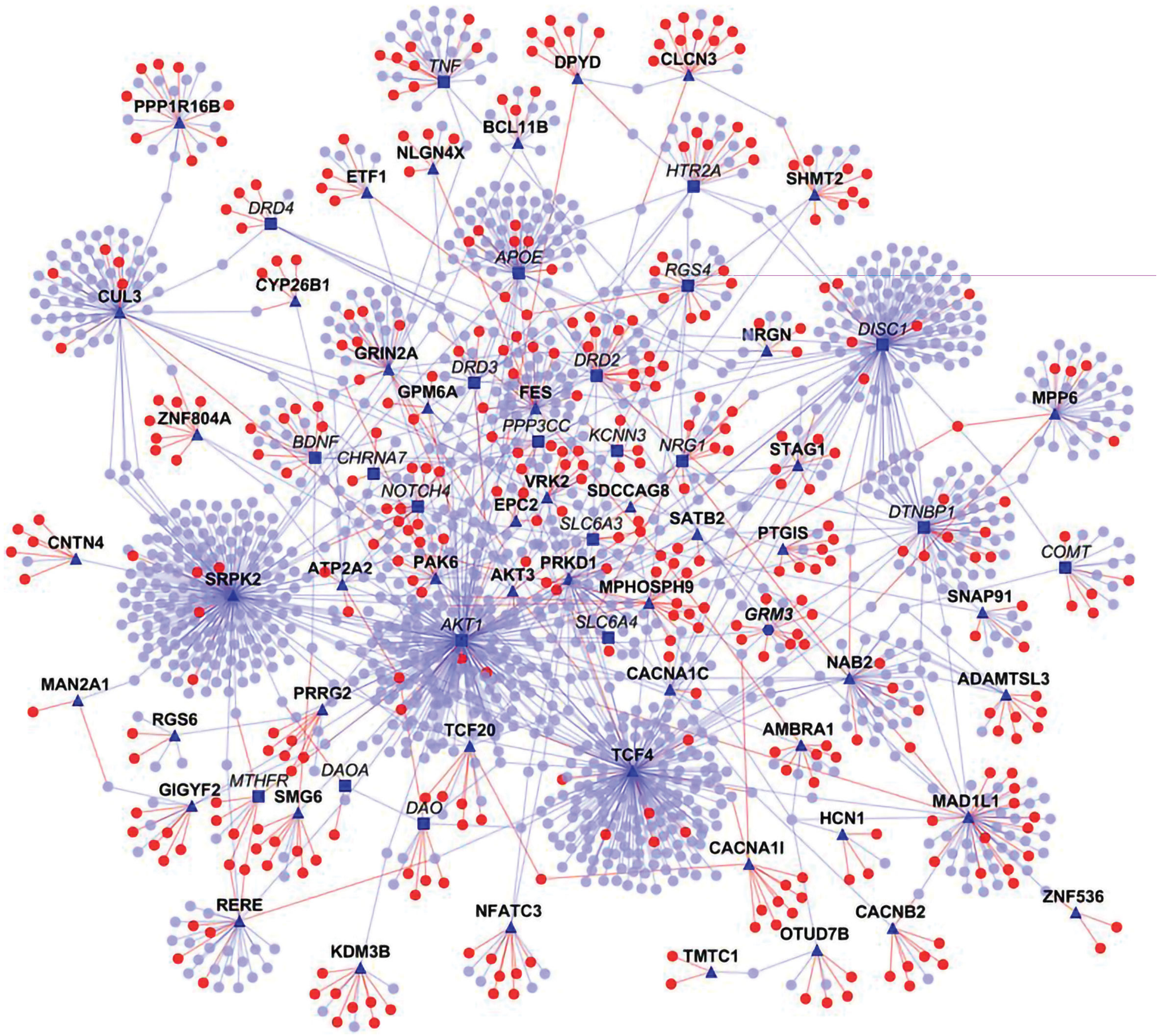
- Relational data
- Knowledge graphs
- Knowledge graph embedding models
- Model expressiveness
- Model inductive capacity & inference patterns
- Empirical evaluation: Datasets and metrics

Overview of the Lecture

- Relational data
- Knowledge graphs
- Knowledge graph embedding models
- Model expressiveness
- Model inductive capacity & inference patterns
- Empirical evaluation: Datasets and metrics
- Summary

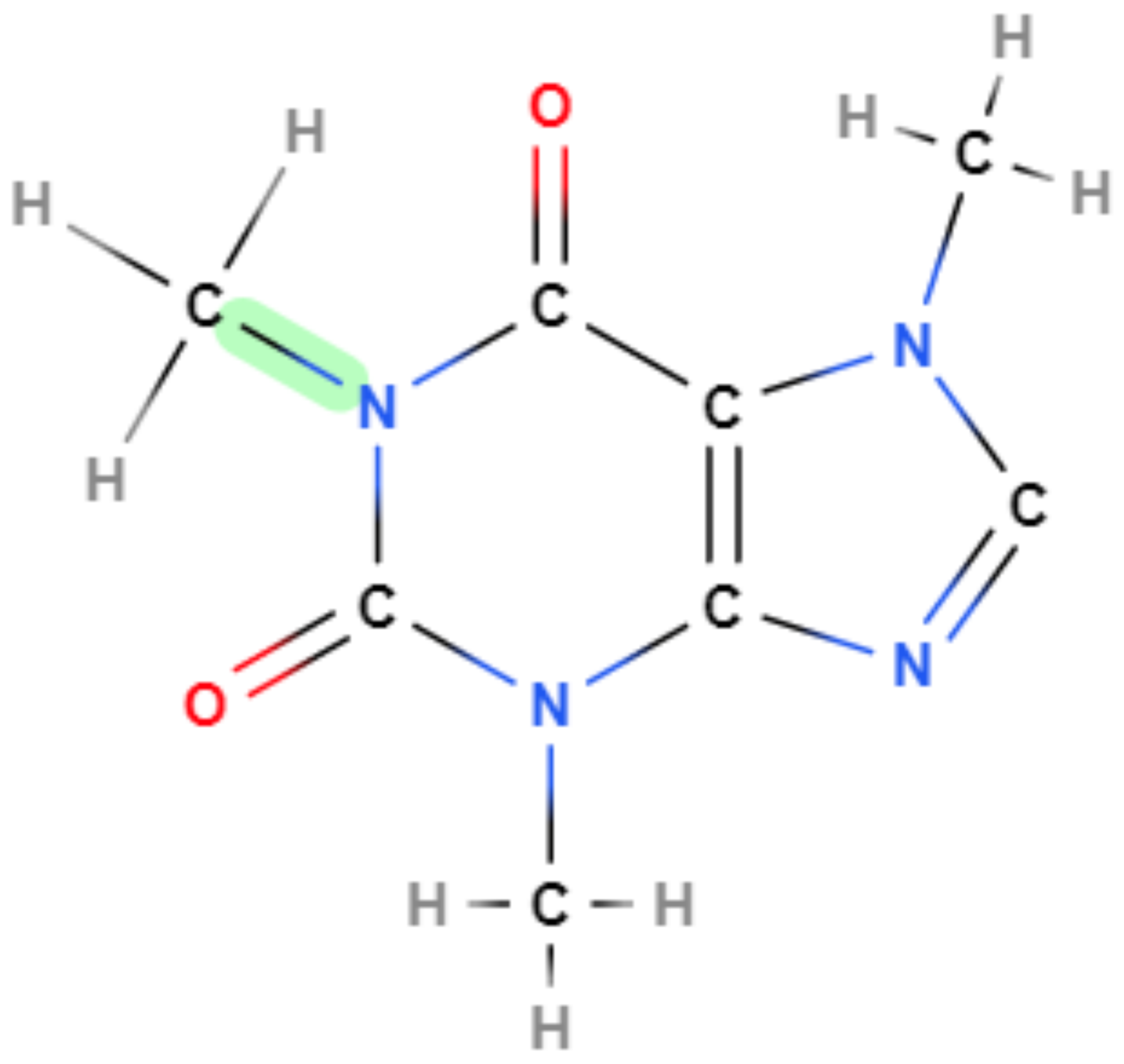
Relational Data

Relational Data



Protein Networks

Relational Data



Molecule Networks

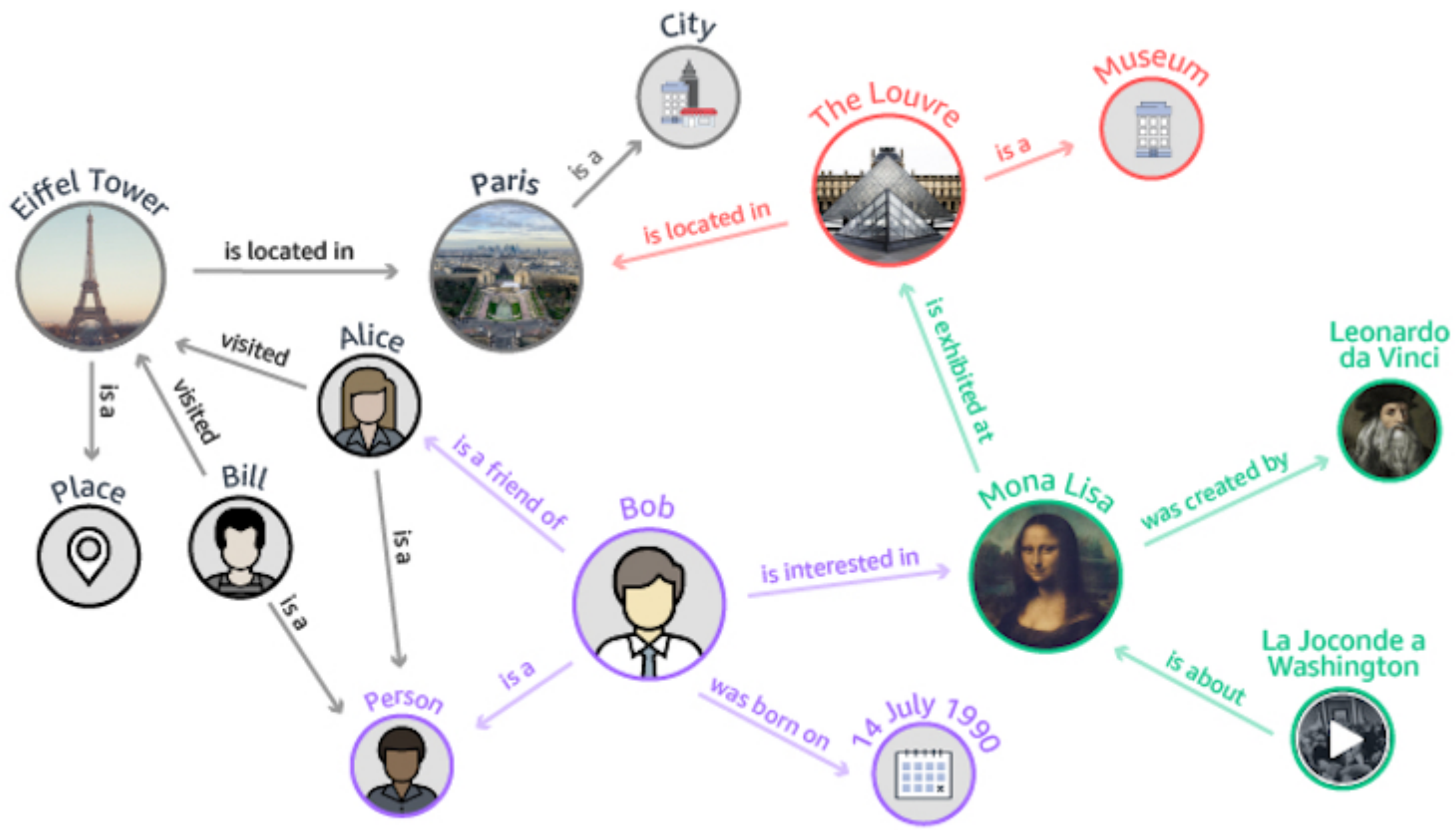
Relational Data

Recommender Systems

Relational Data

Social Networks

Relational Data

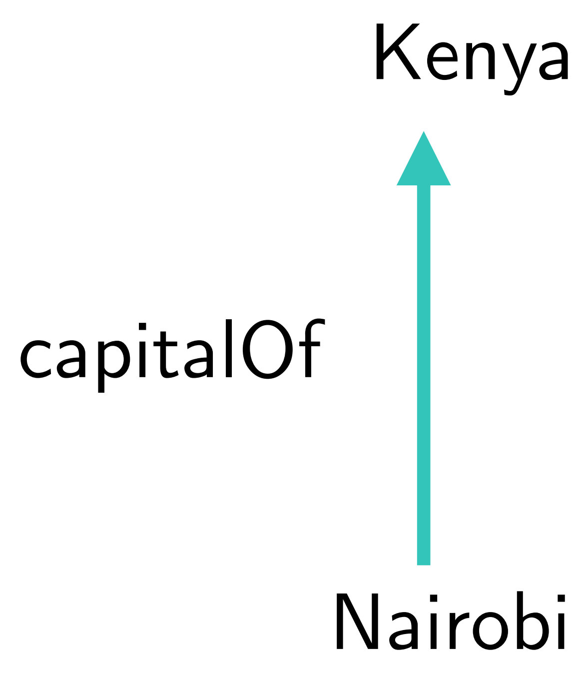


Knowledge Graphs, as graph-structured data models, storing relations (e.g., isFriendOf) between entities (e.g., Alice, Bob) and thereby capture structured knowledge.

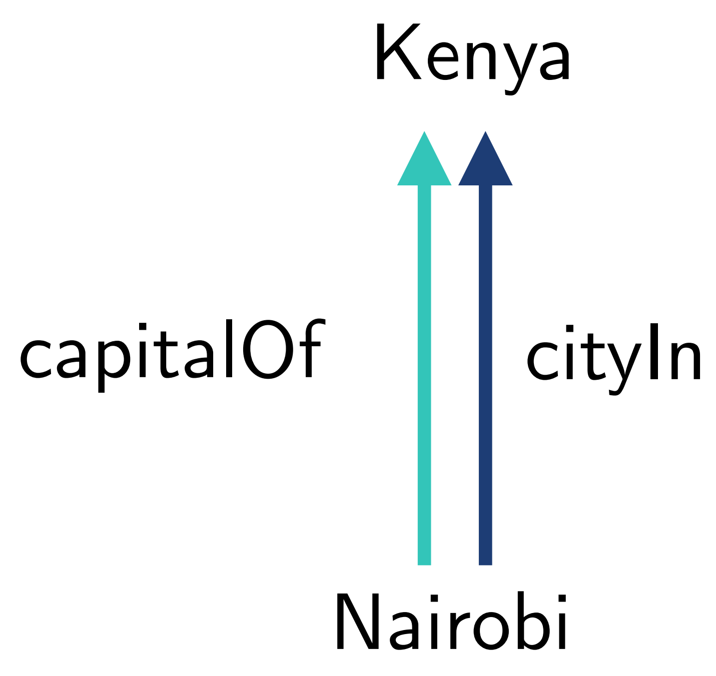
Knowledge Graphs

Knowledge Graphs

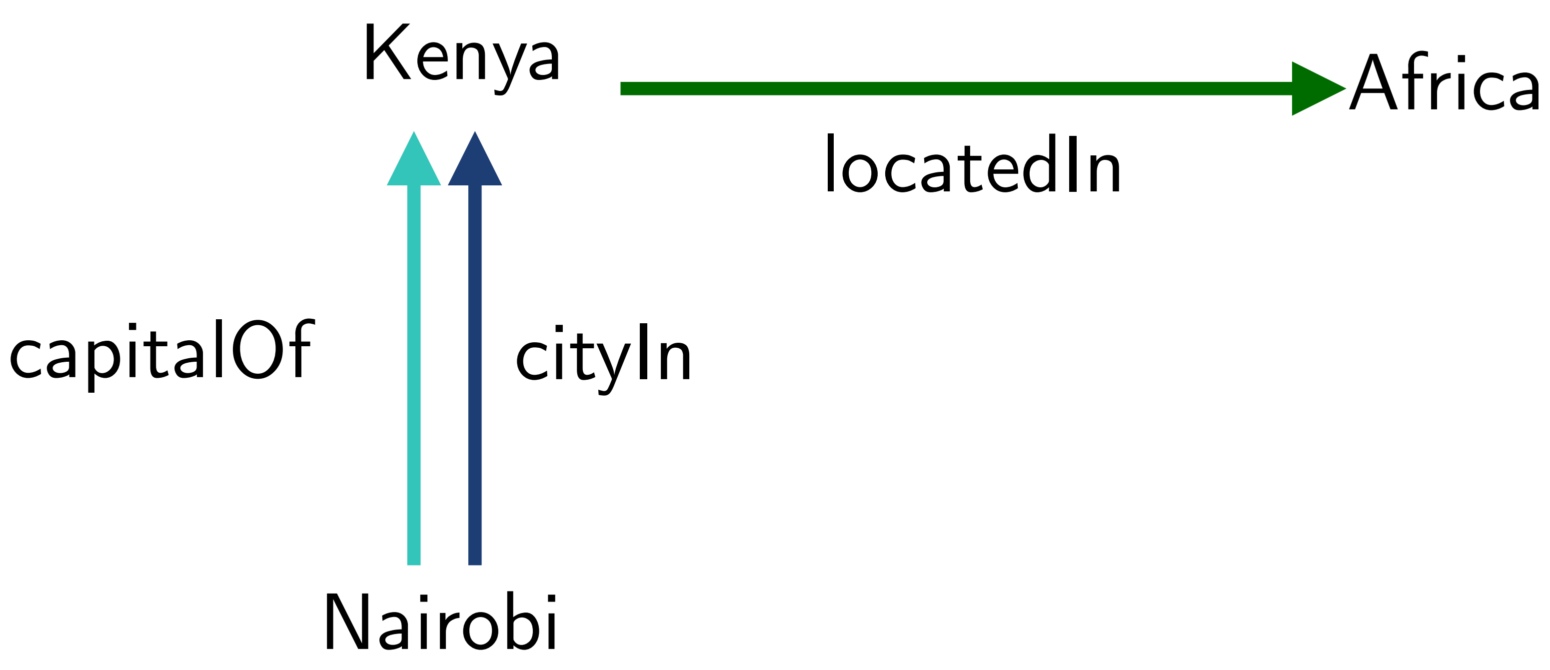
Knowledge Graphs



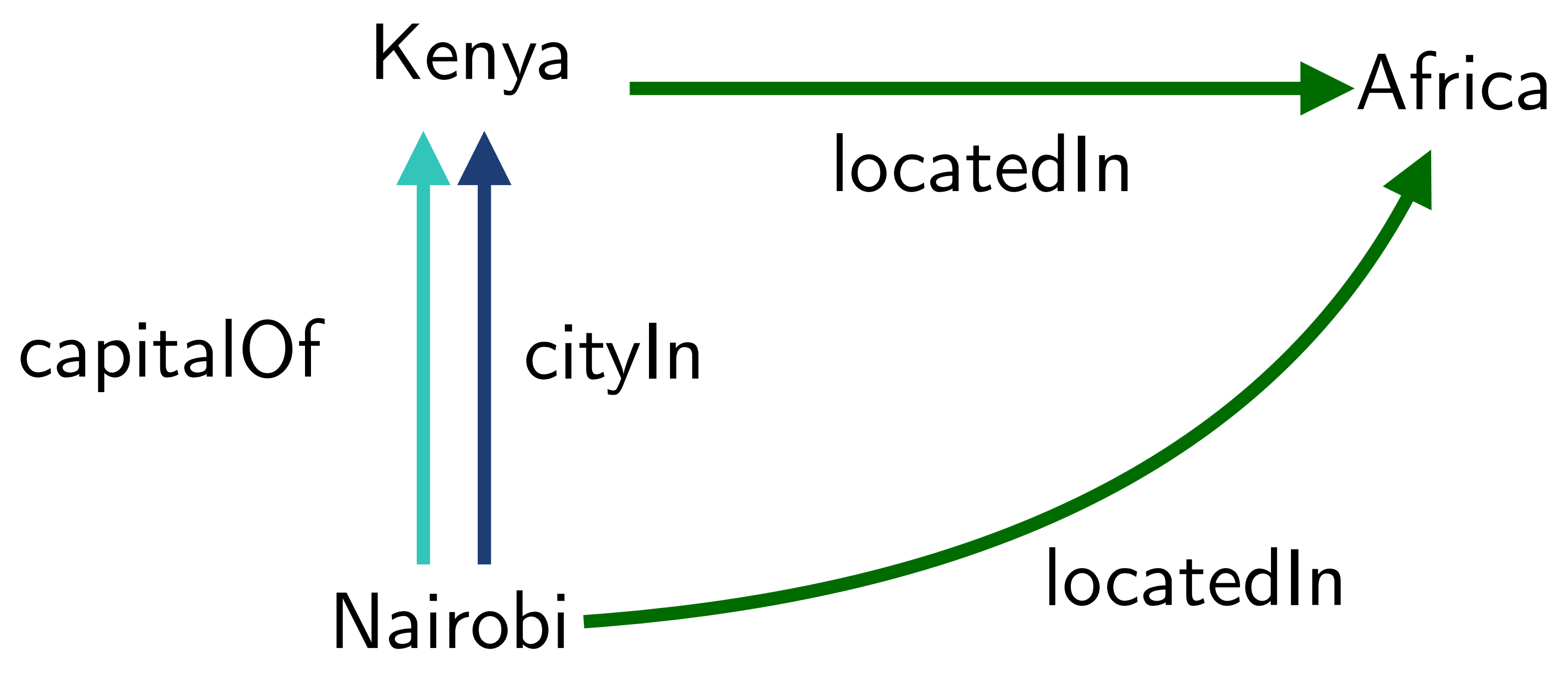
Knowledge Graphs



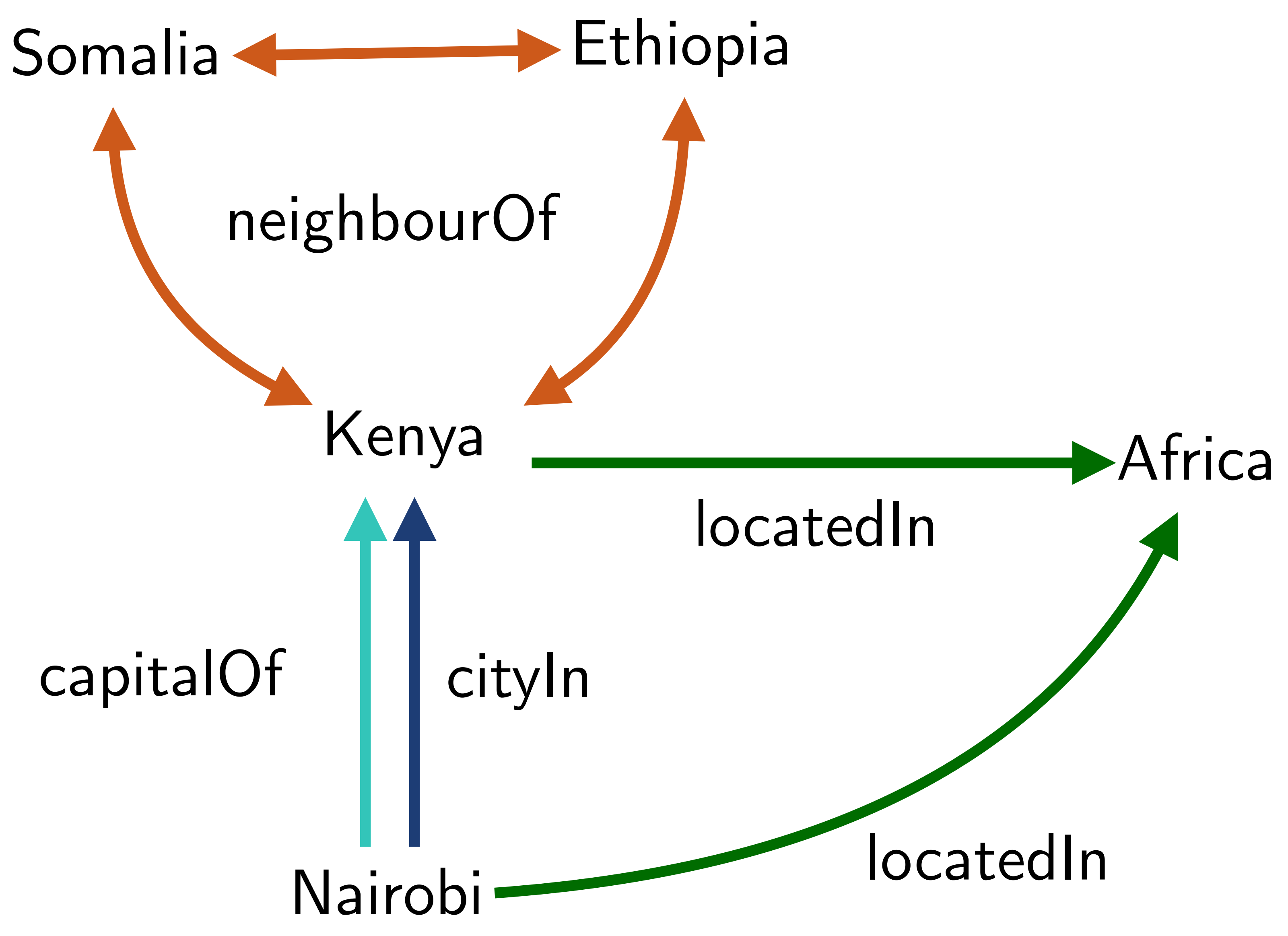
Knowledge Graphs



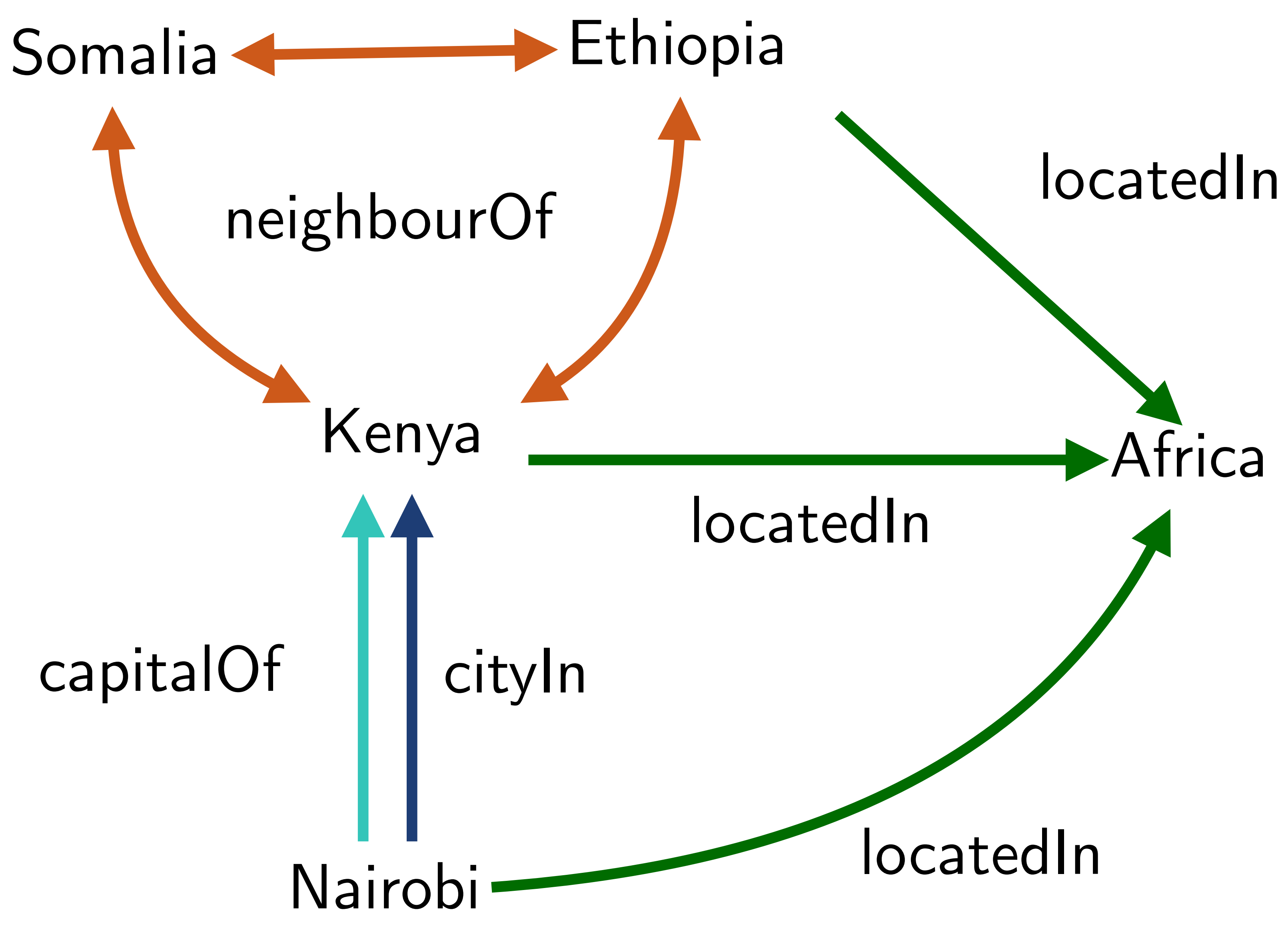
Knowledge Graphs



Knowledge Graphs

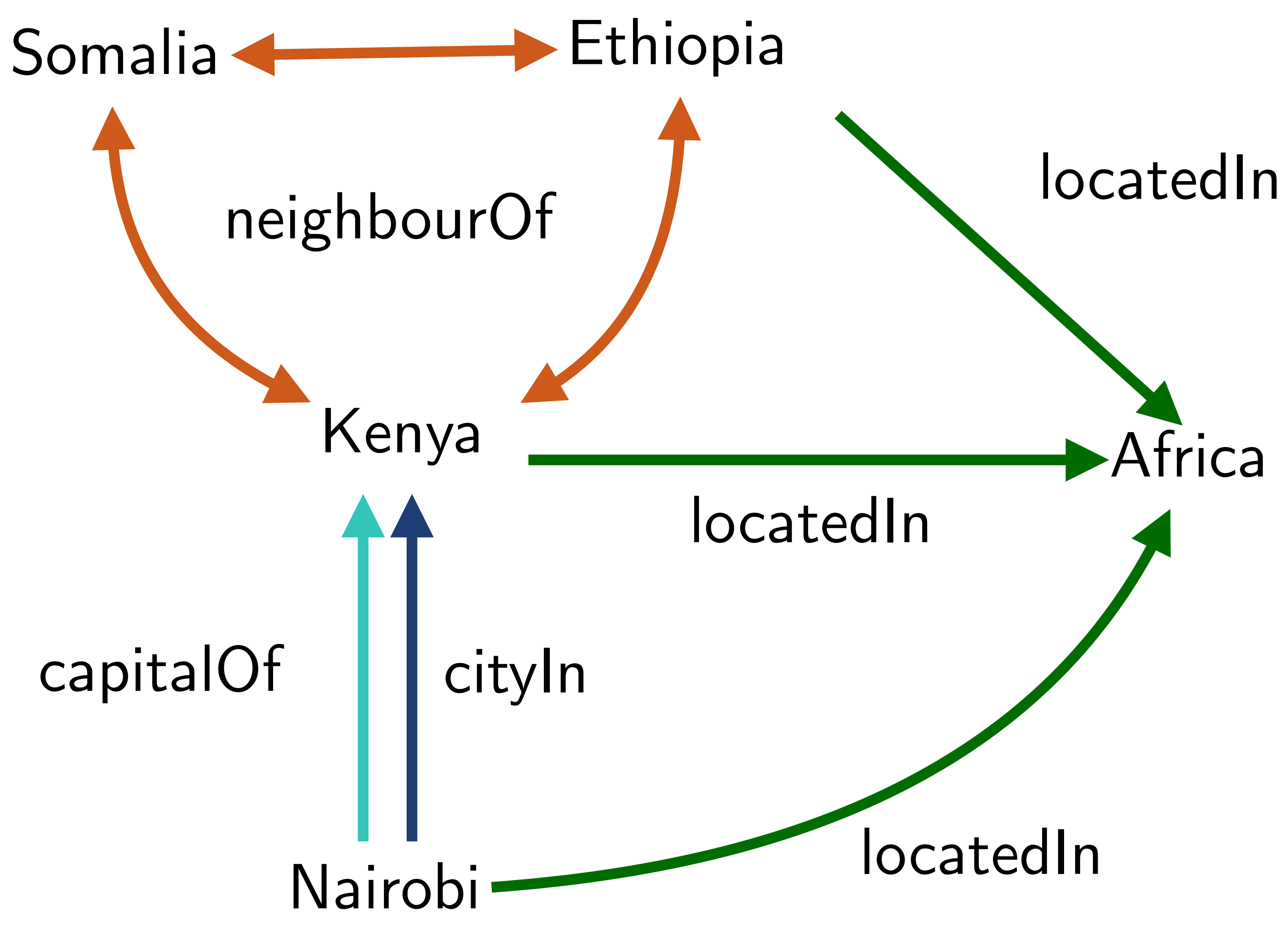


Knowledge Graphs



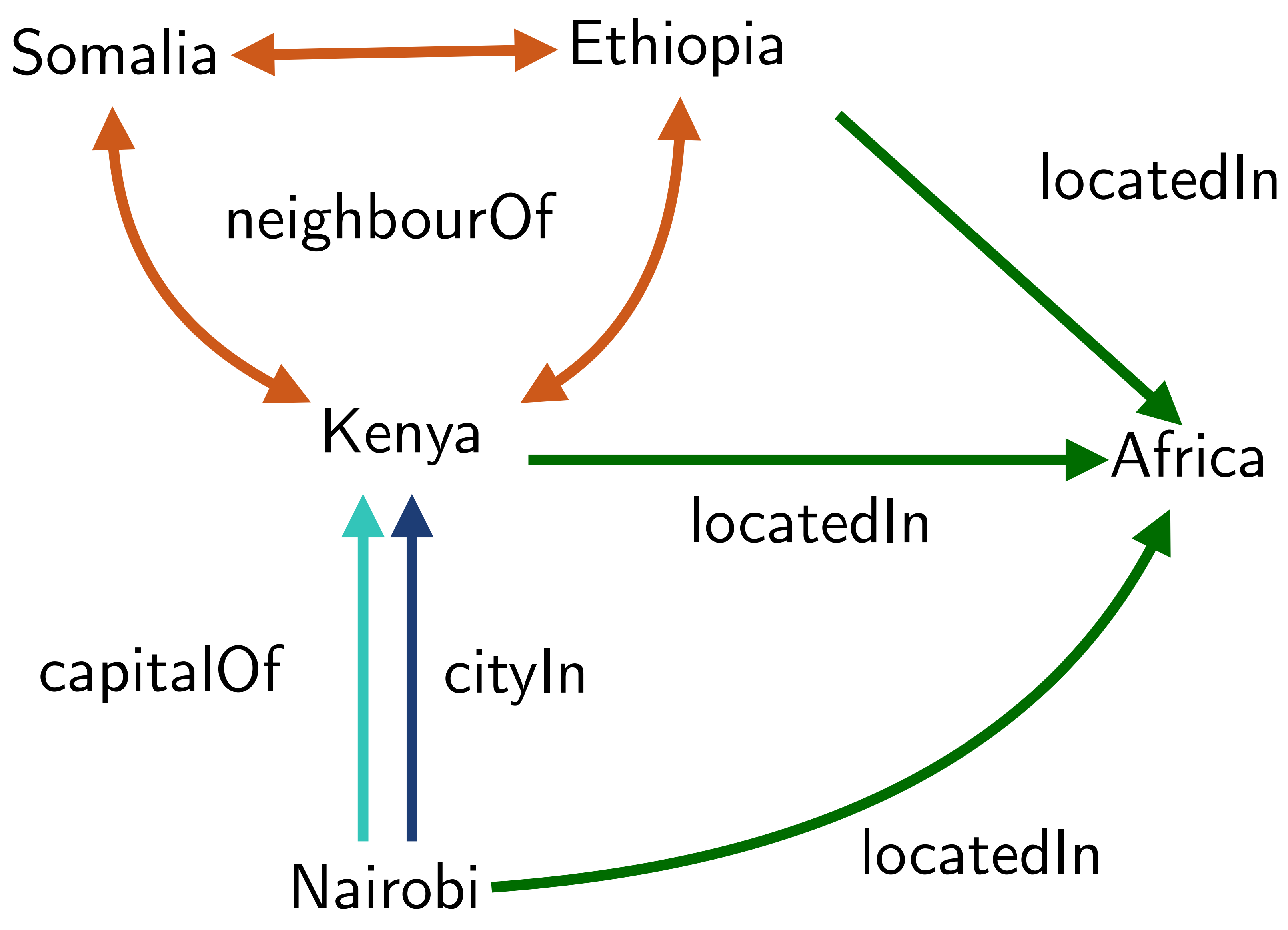
Knowledge Graphs

- We consider a relational vocabulary that consists of a finite set E of entities, and a finite set R of relations.



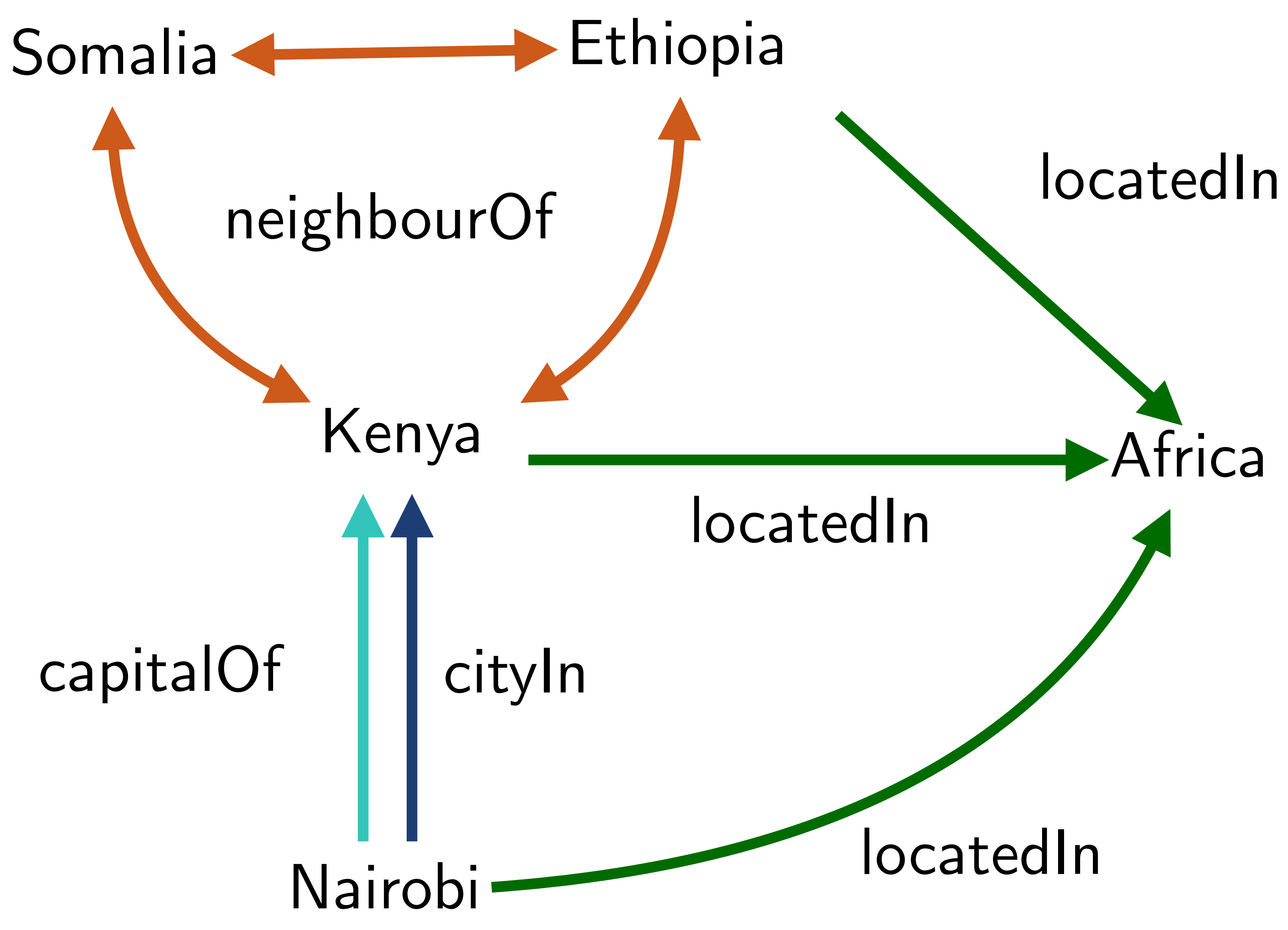
Knowledge Graphs

- We consider a relational vocabulary that consists of a finite set E of entities, and a finite set R of relations.
- A fact is an expression of the form $r(h,t)$, where $r \in R$, and $h, t \in E$.



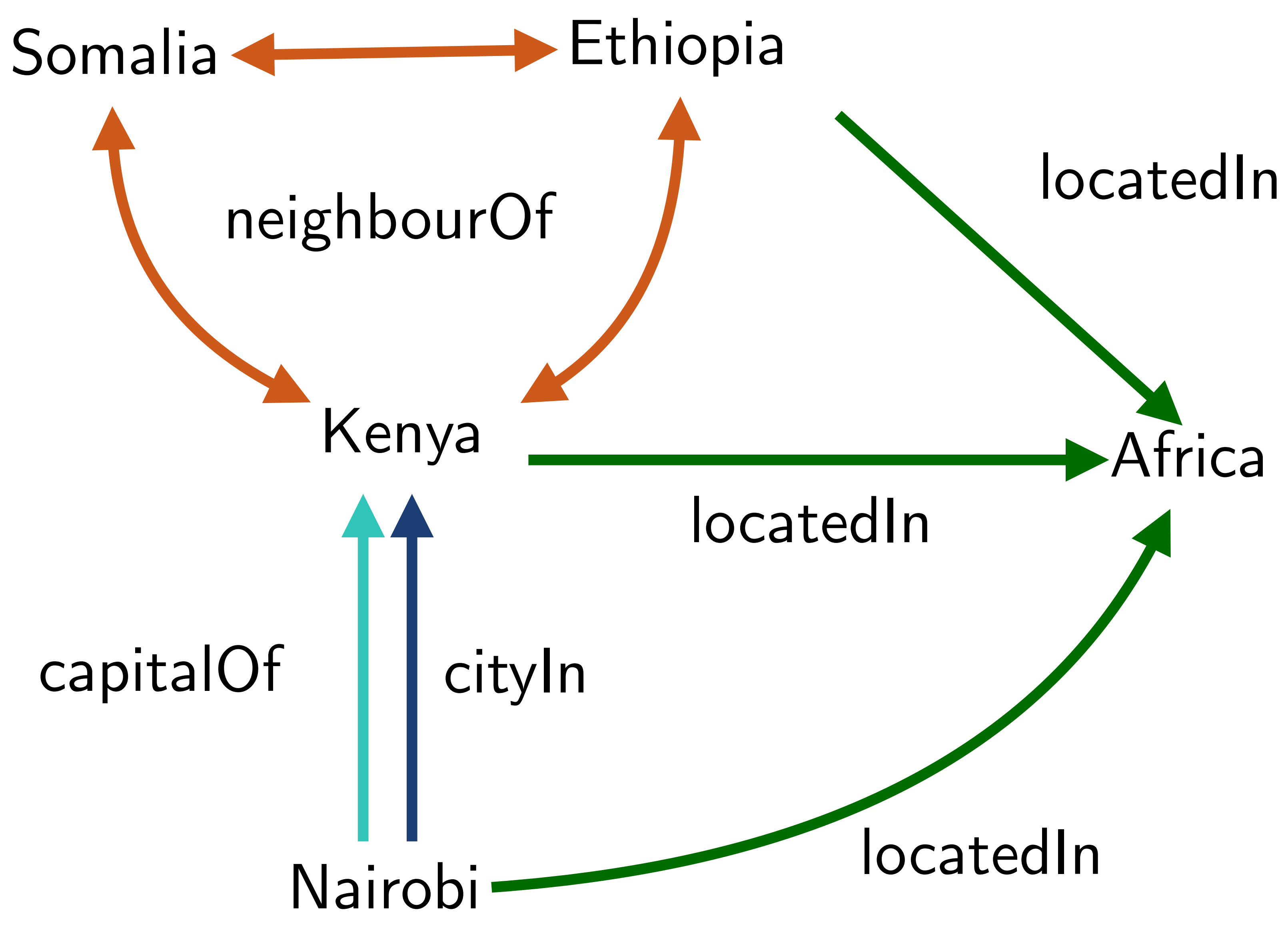
Knowledge Graphs

- We consider a relational vocabulary that consists of a finite set E of entities, and a finite set R of relations.
- A fact is an expression of the form $r(h, t)$, where $r \in R$, and $h, t \in E$.
- Following a common convention, we refer to h as the head and t as the tail entity in a fact $r(h, t)$. In the literature, such facts are sometimes denoted as triples of the form (h, r, t), i.e., as "subject, predicate, object" triples.



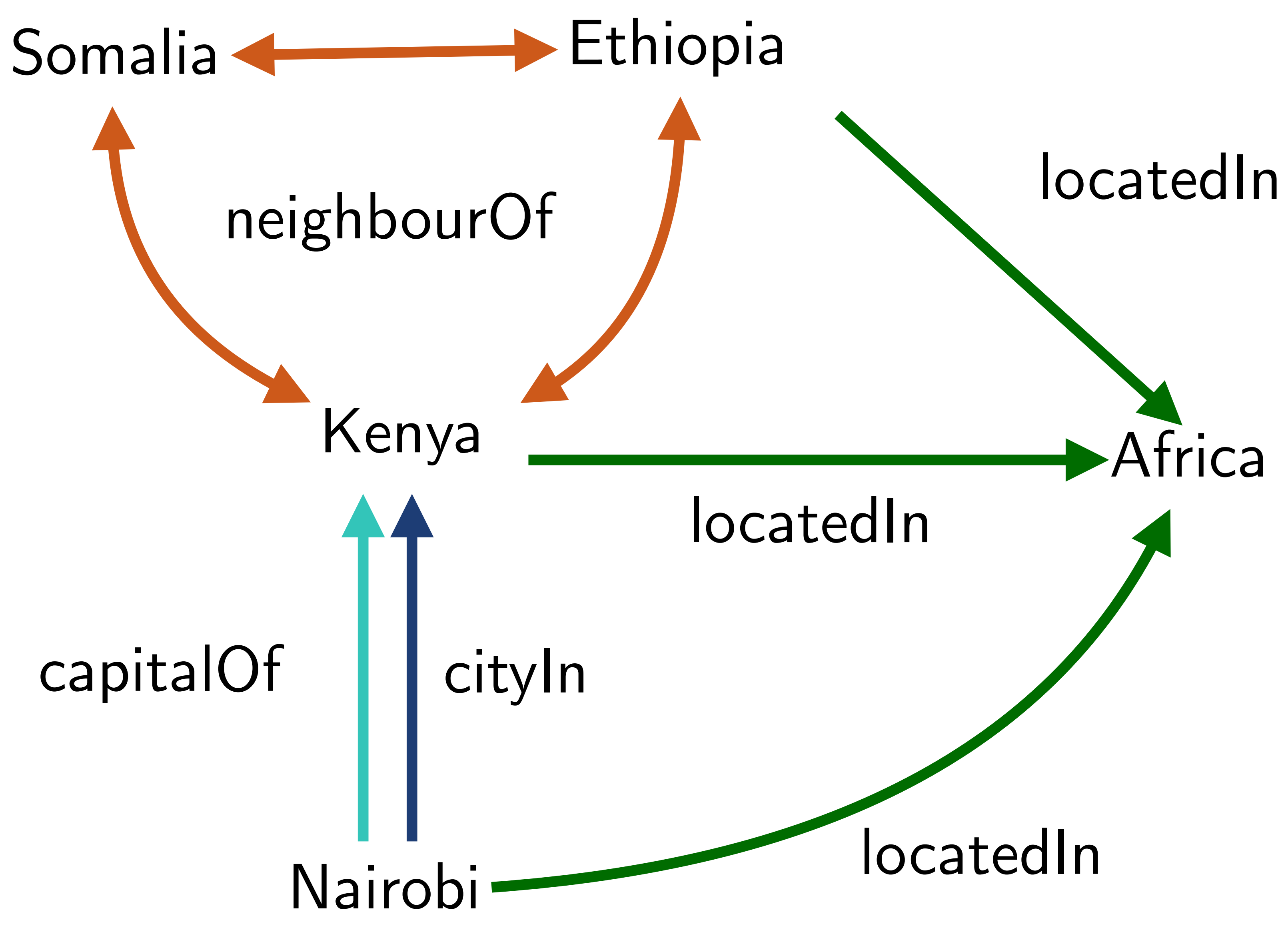
Knowledge Graphs

- We consider a relational vocabulary that consists of a finite set E of entities, and a finite set R of relations.
- A fact is an expression of the form $r(h,t)$, where $r \in R$, and $h,t \in E$.
- Following a common convention, we refer to h as the head and t as the tail entity in a fact $r(h,t)$. In the literature, such facts are sometimes denoted as triples of the form (h,r,t), i.e., as "subject, predicate, object" triples.
- A knowledge graph (KG) G is a set of facts over E and R. Alternatively, we can view a KG as a directed, labelled multigraph $G = (E,R)$ over nodes E and edges R.



Knowledge Graphs

- We consider a relational vocabulary that consists of a finite set E of entities, and a finite set R of relations.
- A fact is an expression of the form $r(h,t)$, where $r \in R$, and $h, t \in E$.
- Following a common convention, we refer to h as the head and t as the tail entity in a fact $r(h,t)$. In the literature, such facts are sometimes denoted as triples of the form (h,r,t), i.e., as "subject, predicate, object" triples.
- A knowledge graph (KG) G is a set of facts over E and R. Alternatively, we can view a KG as a directed, labelled multigraph $G=(E,R)$ over nodes E and edges R.
- We sometimes write U to denote the set of all possible facts over E and R.



Why Knowledge Graphs?

Why Knowledge Graphs?

- KGs provide means for storing, processing, and managing structured data, and are part of modern information technologies.

Why Knowledge Graphs?

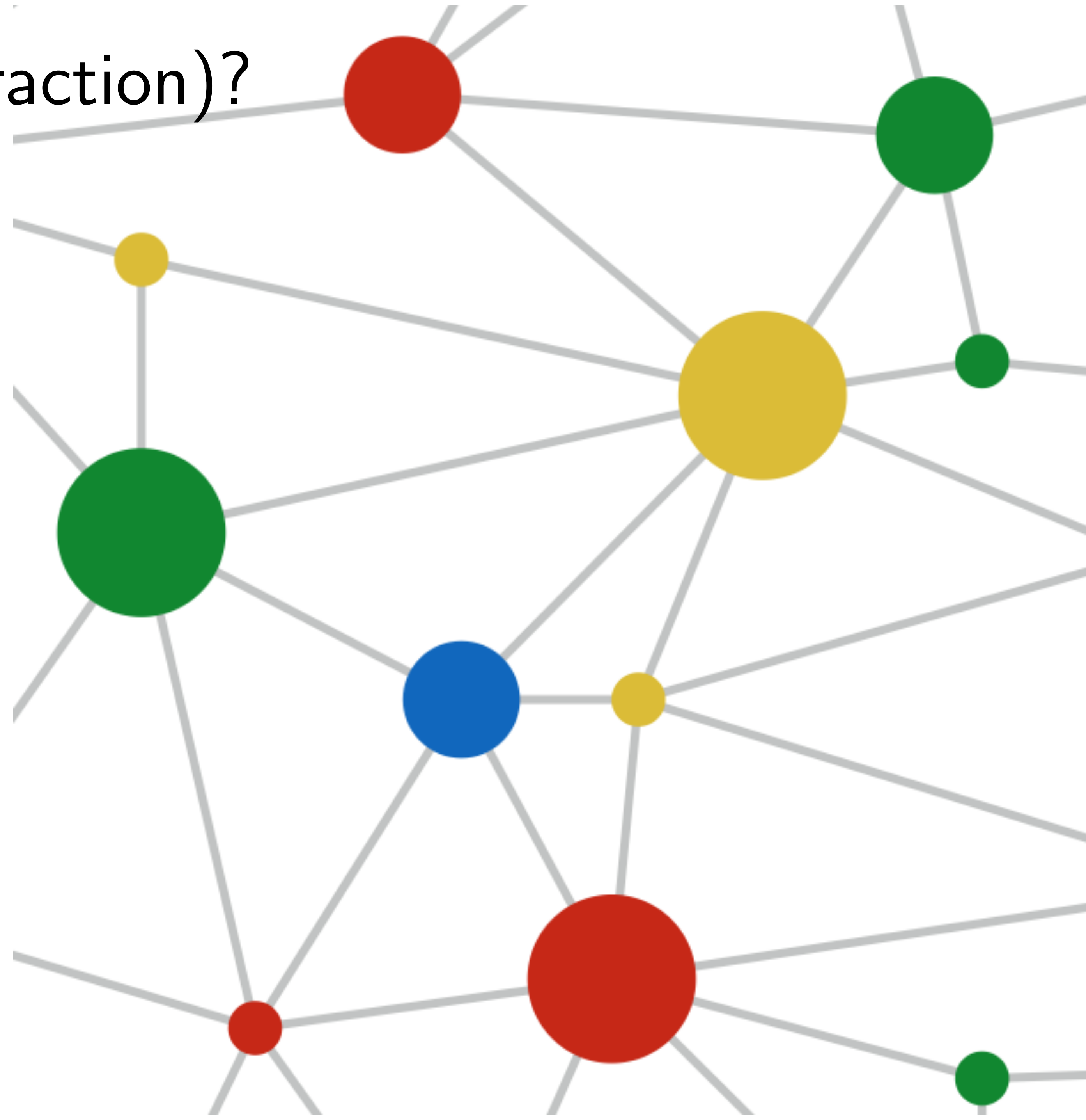
- KGs provide means for storing, processing, and managing structured data, and are part of modern information technologies.
- KGs can be used for reasoning (in conjunction with ontologies), and for query answering, i.e., "Who has co-authored a paper with Marie Curie and Pierre Curie?"

Why Knowledge Graphs?

- KGs provide means for storing, processing, and managing structured data, and are part of modern information technologies.
- KGs can be used for reasoning (in conjunction with ontologies), and for query answering, i.e., "Who has co-authored a paper with Marie Curie and Pierre Curie?"
- KGs pose (or, relate to) various challenges in AI & machine learning:

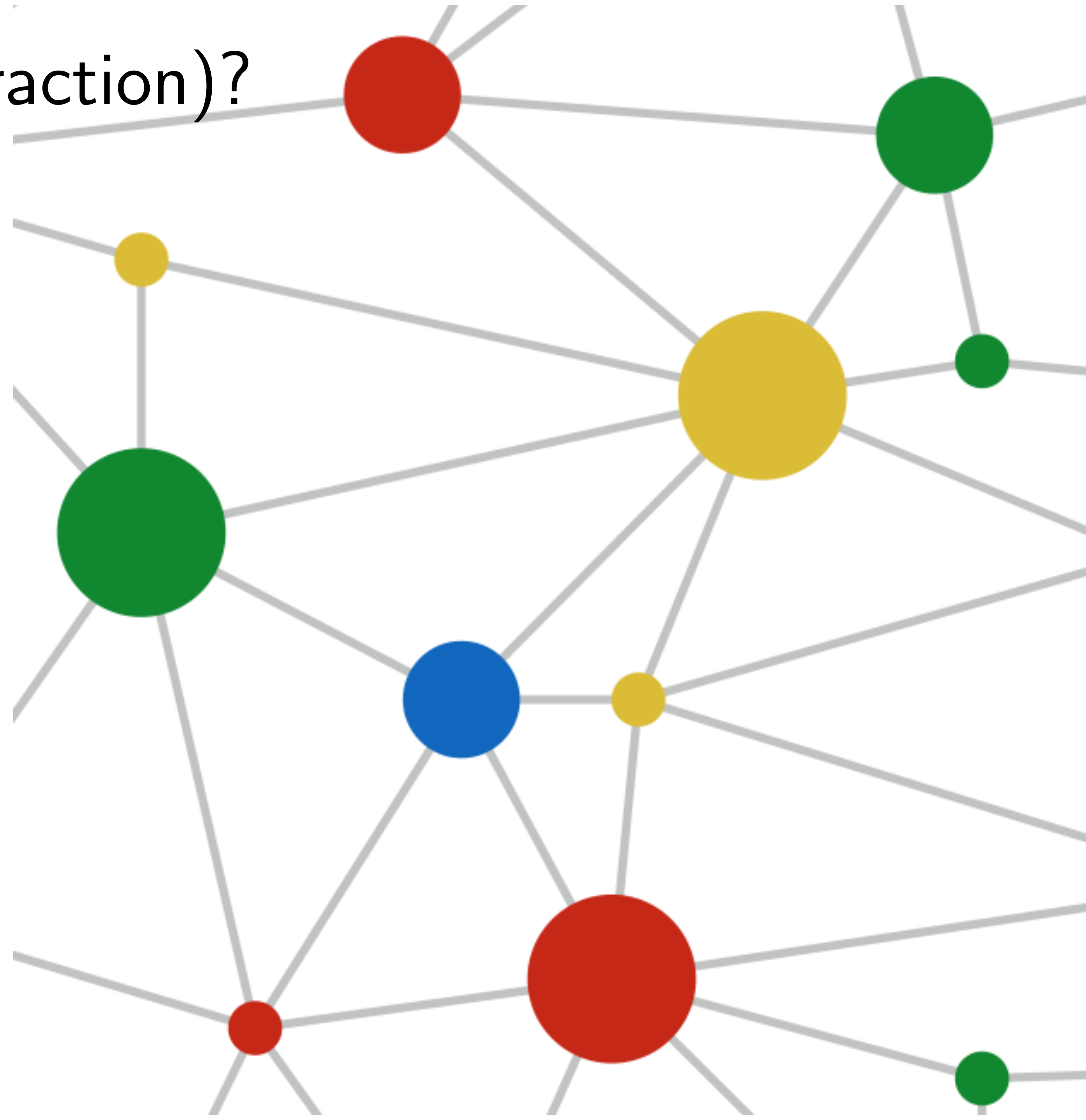
Why Knowledge Graphs?

- KGs provide means for storing, processing, and managing structured data, and are part of modern information technologies.
- KGs can be used for reasoning (in conjunction with ontologies), and for query answering, i.e., "Who has co-authored a paper with Marie Curie and Pierre Curie?"
- KGs pose (or, relate to) various challenges in AI & machine learning:
  - How to automatically construct KGs (e.g., relation extraction, open information extraction)?



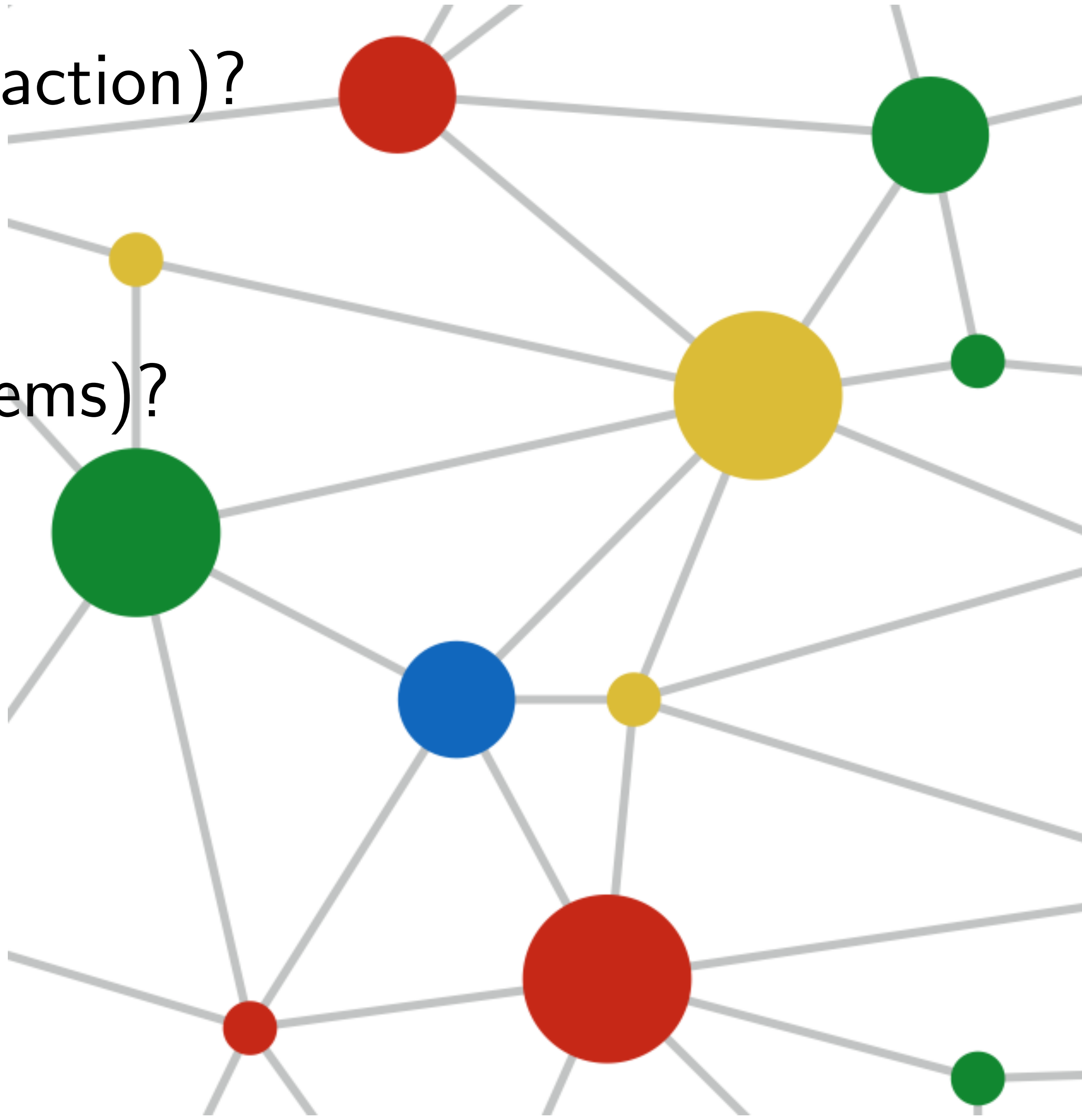
Why Knowledge Graphs?

- KGs provide means for storing, processing, and managing structured data, and are part of modern information technologies.
- KGs can be used for reasoning (in conjunction with ontologies), and for query answering, i.e., "Who has co-authored a paper with Marie Curie and Pierre Curie?"
- KGs pose (or, relate to) various challenges in AI & machine learning:
  - How to automatically construct KGs (e.g., relation extraction, open information extraction)?
  - How to populate an existing KG with new facts (e.g., KG completion)?



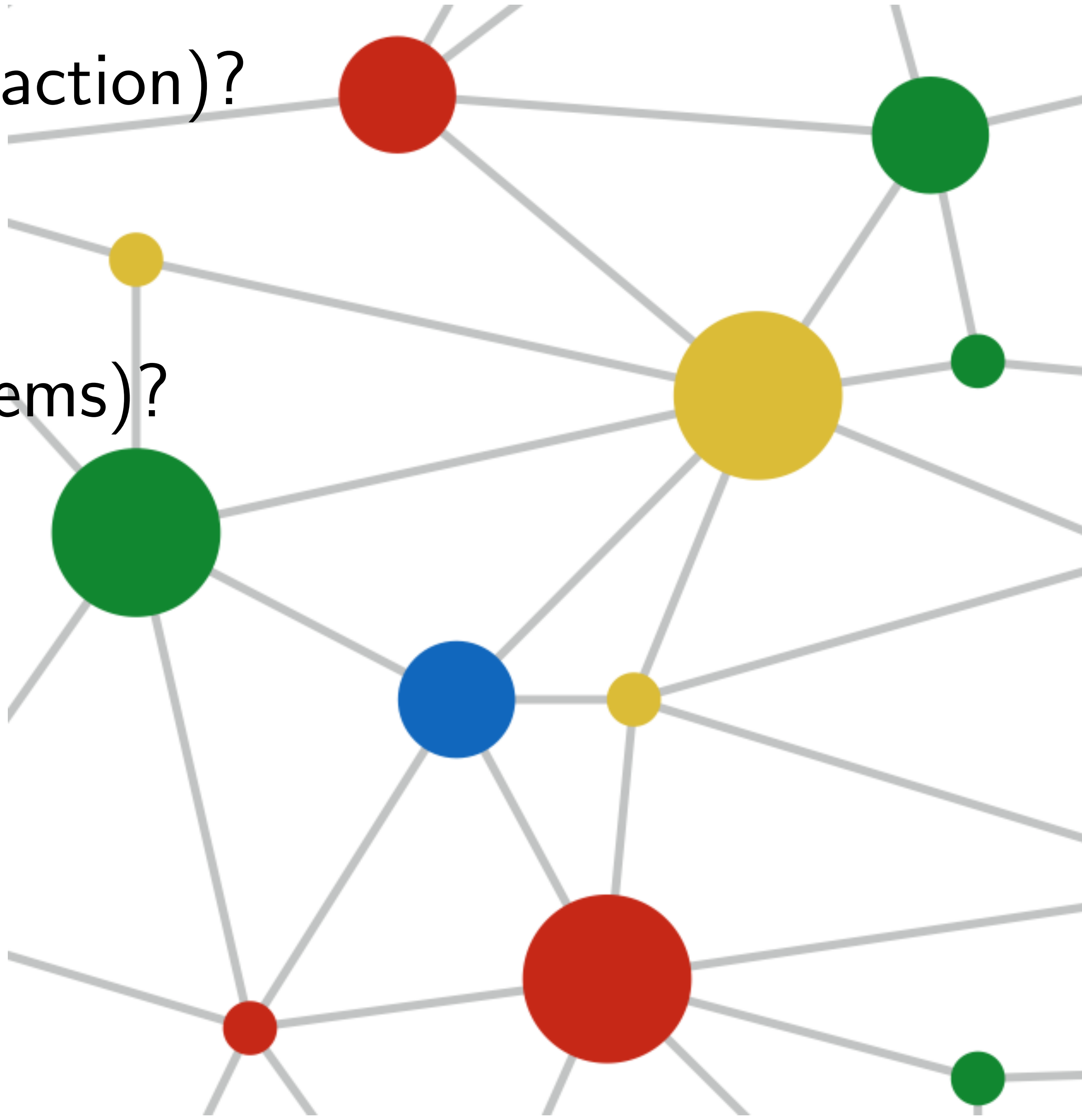
Why Knowledge Graphs?

- KGs provide means for storing, processing, and managing structured data, and are part of modern information technologies.
- KGs can be used for reasoning (in conjunction with ontologies), and for query answering, i.e., "Who has co-authored a paper with Marie Curie and Pierre Curie?"
- KGs pose (or, relate to) various challenges in AI & machine learning:
  - How to automatically construct KGs (e.g., relation extraction, open information extraction)?
  - How to populate an existing KG with new facts (e.g., KG completion)?
  - How to improve/personalise information systems using KGs (e.g., recommender systems)?



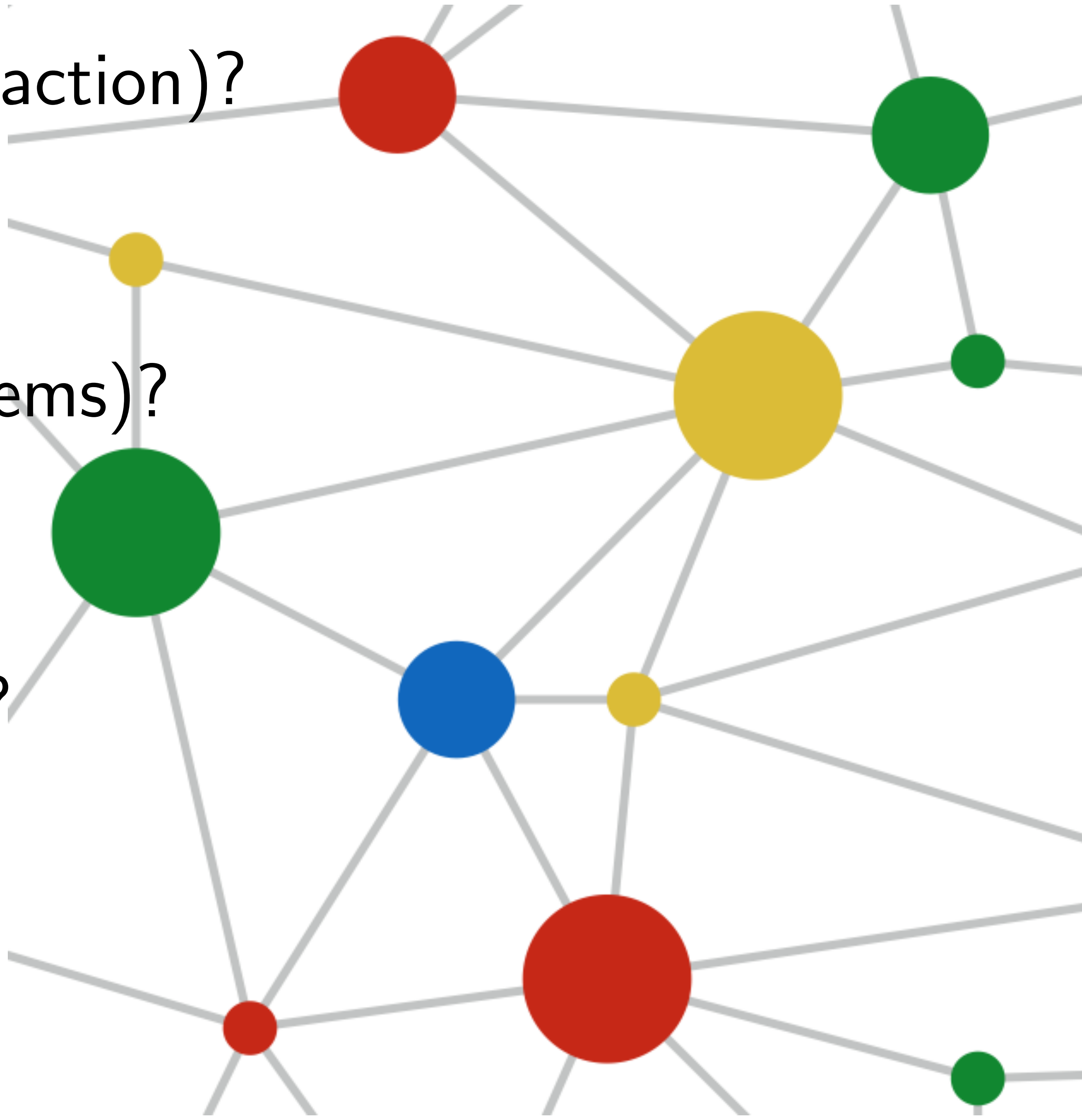
Why Knowledge Graphs?

- KGs provide means for storing, processing, and managing structured data, and are part of modern information technologies.
- KGs can be used for reasoning (in conjunction with ontologies), and for query answering, i.e., “Who has co-authored a paper with Marie Curie and Pierre Curie?”
- KGs pose (or, relate to) various challenges in AI & machine learning:
  - How to automatically construct KGs (e.g., relation extraction, open information extraction)?
  - How to populate an existing KG with new facts (e.g., KG completion)?
  - How to improve/personalise information systems using KGs (e.g., recommender systems)?
  - How to learn on top of KGs, while complying with the existing knowledge?



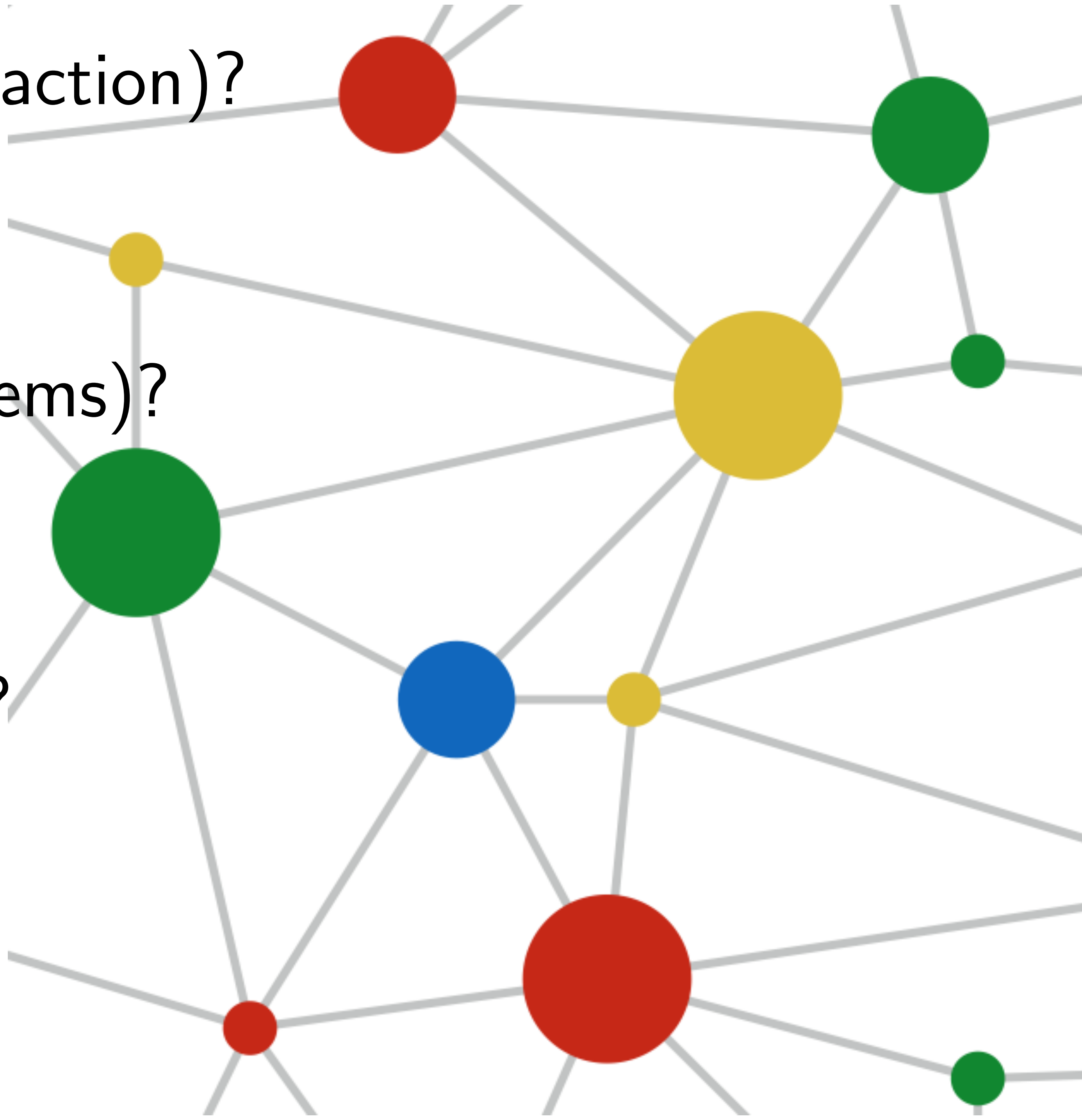
Why Knowledge Graphs?

- KGs provide means for storing, processing, and managing structured data, and are part of modern information technologies.
- KGs can be used for reasoning (in conjunction with ontologies), and for query answering, i.e., "Who has co-authored a paper with Marie Curie and Pierre Curie?"
- KGs pose (or, relate to) various challenges in AI & machine learning:
  - How to automatically construct KGs (e.g., relation extraction, open information extraction)?
  - How to populate an existing KG with new facts (e.g., KG completion)?
  - How to improve/personalise information systems using KGs (e.g., recommender systems)?
  - How to learn on top of KGs, while complying with the existing knowledge?
  - Can KGs be mediators for developing more reliable and interpretable models for ML?



Why Knowledge Graphs?

- KGs provide means for storing, processing, and managing structured data, and are part of modern information technologies.
- KGs can be used for reasoning (in conjunction with ontologies), and for query answering, i.e., "Who has co-authored a paper with Marie Curie and Pierre Curie?"
- KGs pose (or, relate to) various challenges in AI & machine learning:
  - How to automatically construct KGs (e.g., relation extraction, open information extraction)?
  - How to populate an existing KG with new facts (e.g., KG completion)?
  - How to improve/personalise information systems using KGs (e.g., recommender systems)?
  - How to learn on top of KGs, while complying with the existing knowledge?
  - Can KGs be mediators for developing more reliable and interpretable models for ML?
  - How to make learning and reasoning compatible?



Knowledge Graph Completion

Knowledge Graph Completion

Problem: KGs are typically highly incomplete, which makes their downstream use more challenging. For example, 71% of individuals in Freebase lack a connection to a place of birth.

Knowledge Graph Completion

Problem: KGs are typically highly incomplete, which makes their downstream use more challenging. For example, 71% of individuals in Freebase lack a connection to a place of birth.

Question: Can we automatically find new facts for our KG, solely based on the existing information in the KG?

Knowledge Graph Completion

Problem: KGs are typically highly incomplete, which makes their downstream use more challenging. For example, 71% of individuals in Freebase lack a connection to a place of birth.

Question: Can we automatically find new facts for our KG, solely based on the existing information in the KG?

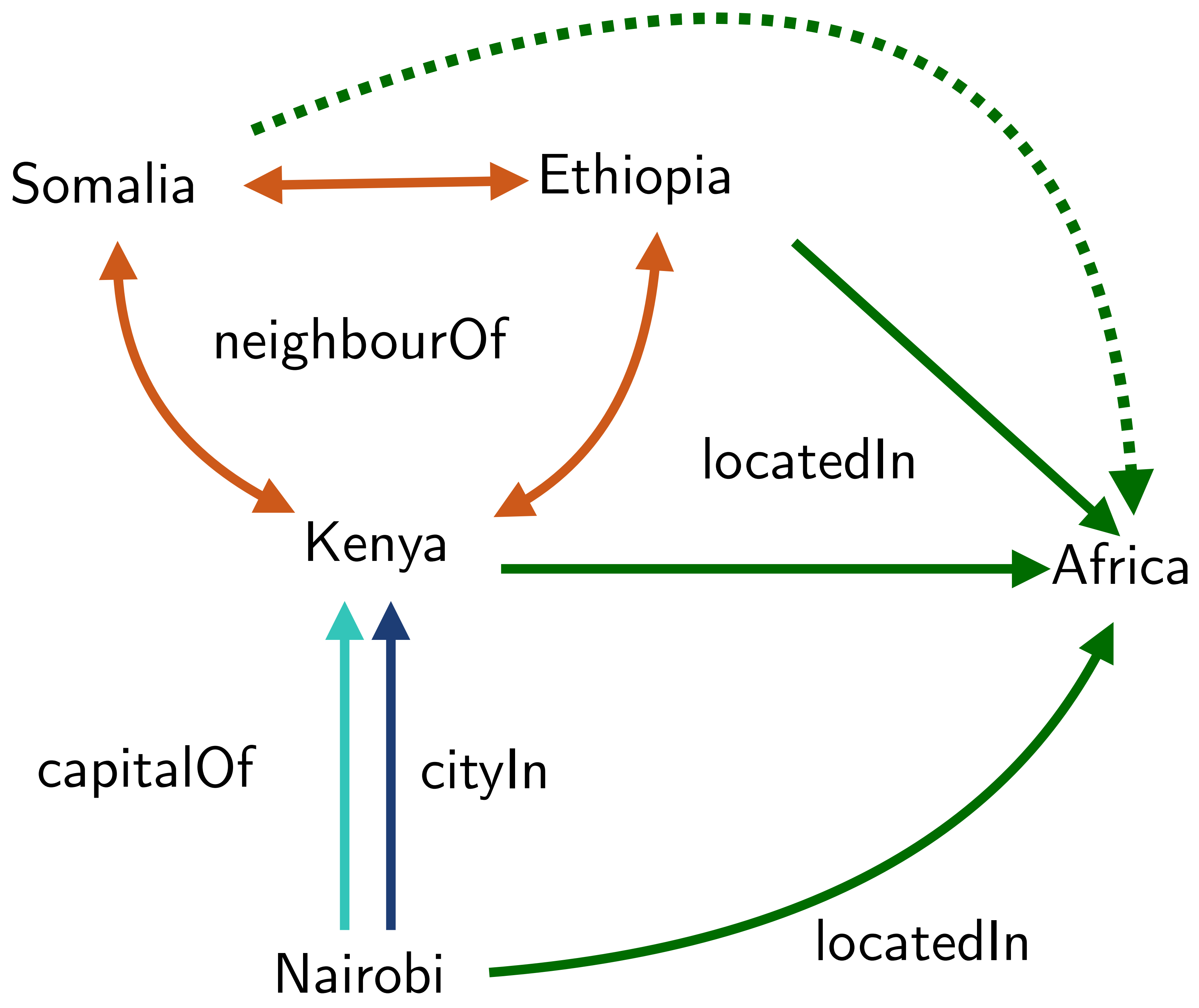
Task: Given a KG G, the task of knowledge graph completion is to predict facts that are missing from G.

Knowledge Graph Completion

Problem: KGs are typically highly incomplete, which makes their downstream use more challenging. For example, 71% of individuals in Freebase lack a connection to a place of birth.

Question: Can we automatically find new facts for our KG, solely based on the existing information in the KG?

Task: Given a KG G, the task of knowledge graph completion is to predict facts that are missing from G.



Inspiration from Word Vector Representations

"The word representations computed using neural networks are very interesting because the learned vectors explicitly encode many linguistic regularities and patterns.

Somewhat surprisingly, many of these patterns can be represented as linear translations.

For example, the result of a vector calculation vec("Madrid") - vec("Spain") + vec("France") is closer to vec("Paris") than to any other word vector."

(Mikolov et. al, 2013)

Inspiration from Word Vector Representations

"The word representations computed using neural networks are very interesting because the learned vectors explicitly encode many linguistic regularities and patterns.

Somewhat surprisingly, many of these patterns can be represented as linear translations.

For example, the result of a vector calculation vec("Madrid") - vec("Spain") + vec("France") is closer to vec("Paris") than to any other word vector."

(Mikolov et. al, 2013)

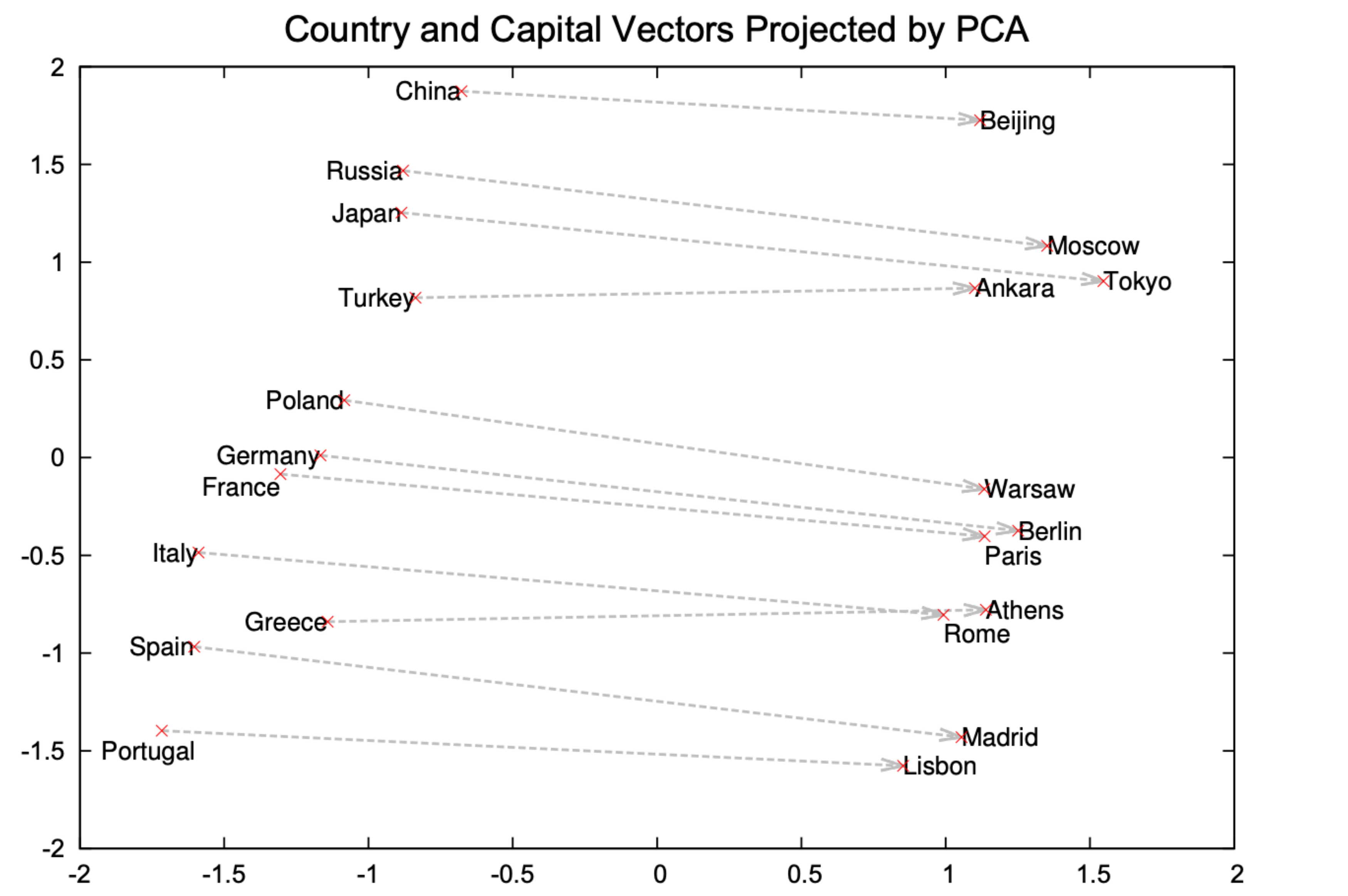


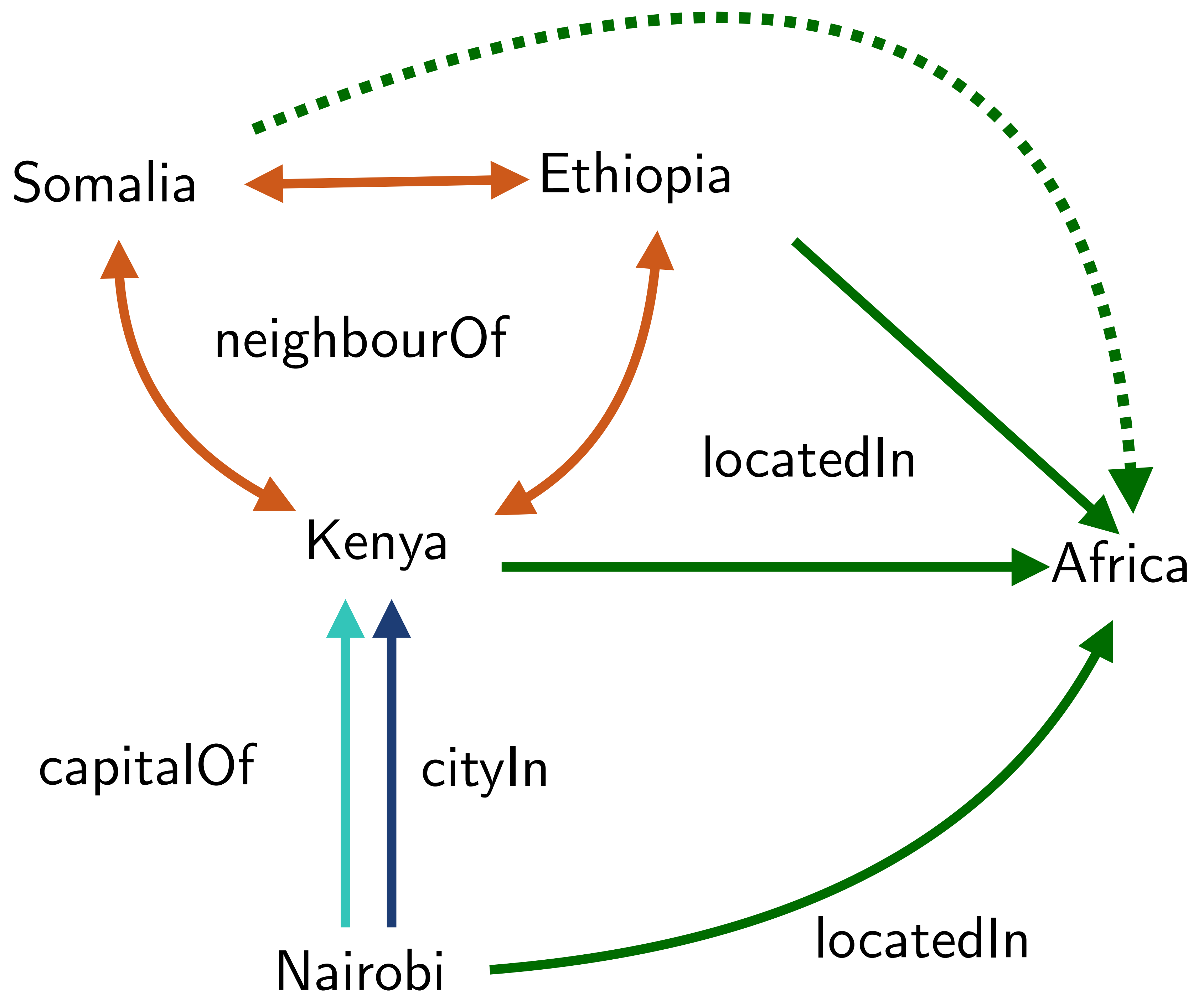
Figure 2 (Mikolov et. al, 2013): 2-dimensional PCA projection of the 1000-dimensional Skip-gram vectors of countries and their capital cities. The figure illustrates ability of the model to automatically organize concepts and learn implicitly the relationships between them, as during the training no supervised information about what a capital city means is given.

Knowledge Graph Completion

Problem: KGs are typically highly incomplete, which makes their downstream use more challenging. For example, 71% of individuals in Freebase lack a connection to a place of birth.

Question: Can we automatically find new facts for our KG, solely based on the existing information in the KG?

Task: Given a KG G, the task of knowledge graph completion is to predict facts that are missing from G.

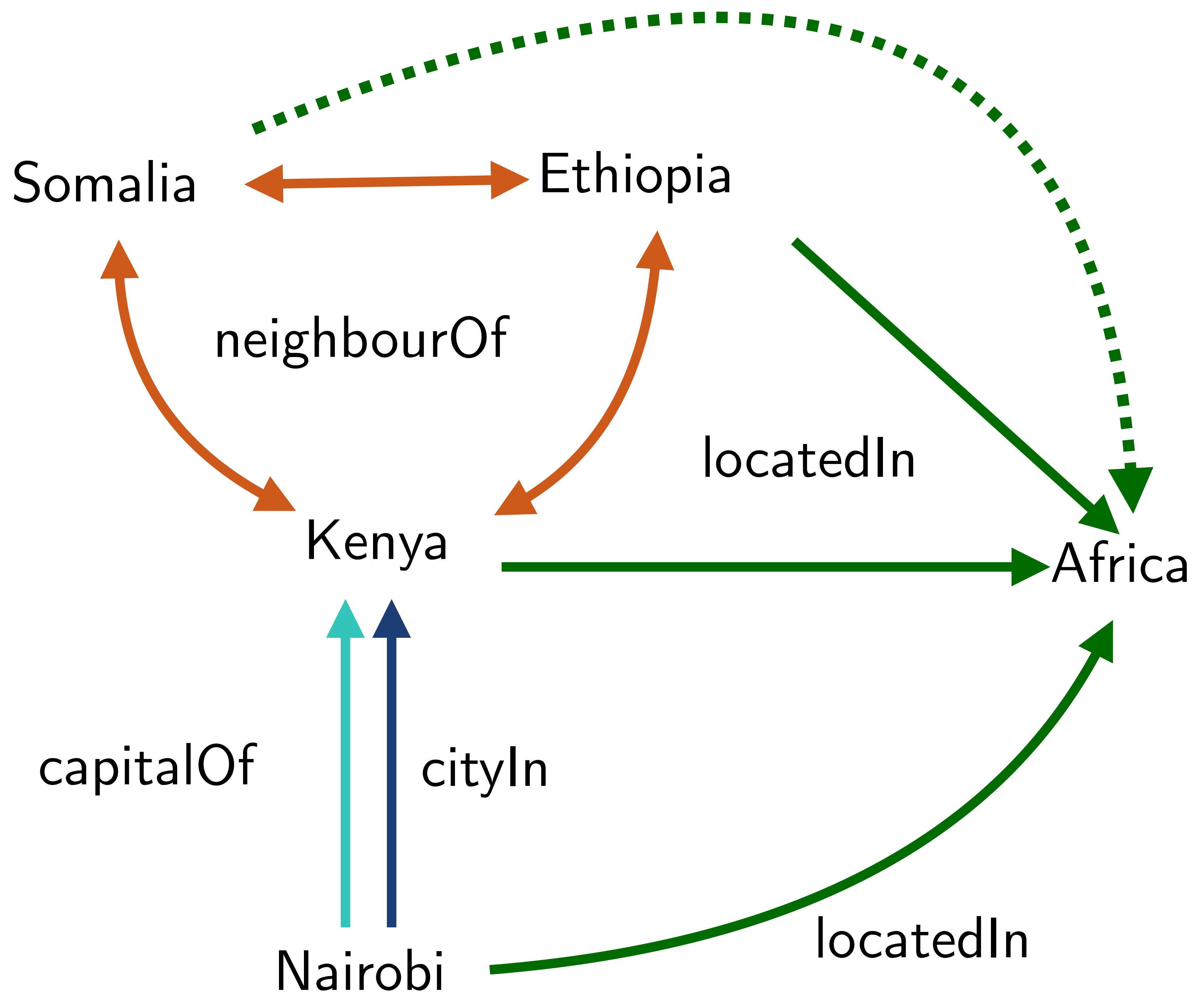


Knowledge Graph Completion

Problem: KGs are typically highly incomplete, which makes their downstream use more challenging. For example, 71% of individuals in Freebase lack a connection to a place of birth.

Question: Can we automatically find new facts for our KG, solely based on the existing information in the KG?

Task: Given a KG G, the task of knowledge graph completion is to predict facts that are missing from G.



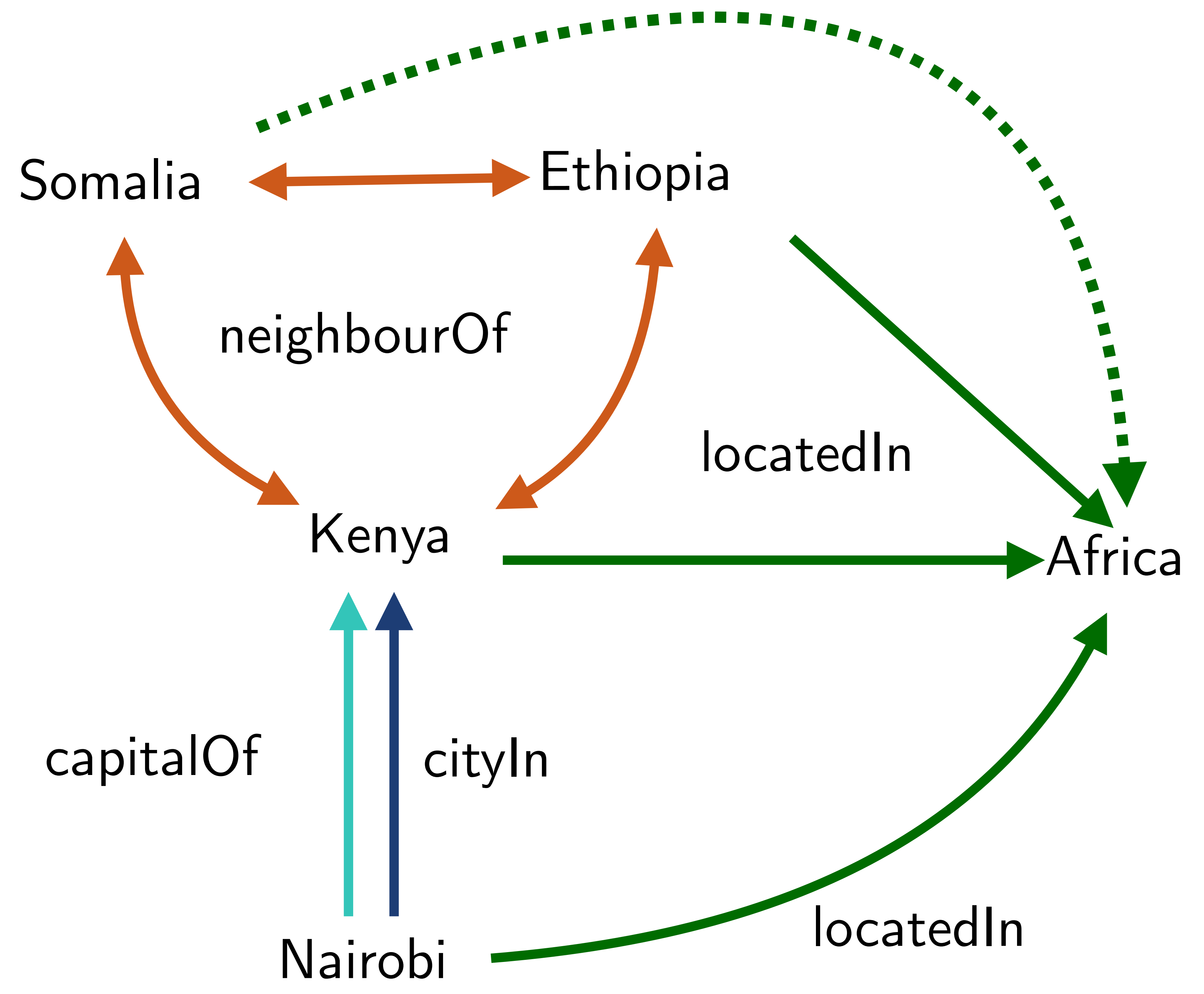
Intuition: Real-world data lies in low dimensional manifolds, so if existing facts in a KG exhibit common patterns then one can embed them into low-dimensional vector-spaces and use them to predict new facts.

Knowledge Graph Completion

Problem: KGs are typically highly incomplete, which makes their downstream use more challenging. For example, 71% of individuals in Freebase lack a connection to a place of birth.

Question: Can we automatically find new facts for our KG, solely based on the existing information in the KG?

Task: Given a KG G, the task of knowledge graph completion is to predict facts that are missing from G.



Intuition: Real-world data lies in low dimensional manifolds, so if existing facts in a KG exhibit common patterns then one can embed them into low-dimensional vector-spaces and use them to predict new facts.

Idea: Represent entities and relations as embeddings, while capturing latent properties of the knowledge graph, i.e., similar entities and relationships will be represented with similar embeddings. Use such similarities to rank new predictions.

Knowledge Graph Embedding Models

KG Embedding Models

KG Embedding Models

Most of the existing approaches can be described in term of the following criteria:

(i) **Model representation**: How are the entities and relations represented?

(ii) **Scoring function**: How is the likelihood of a fact to be true defined?

(iii) **Loss function**: What is the objective function to be minimised?

KG Embedding Models

Most of the existing approaches can be described in term of the following criteria:

(i) **Model representation**: How are the entities and relations represented?

(ii) **Scoring function**: How is the likelihood of a fact to be true defined?

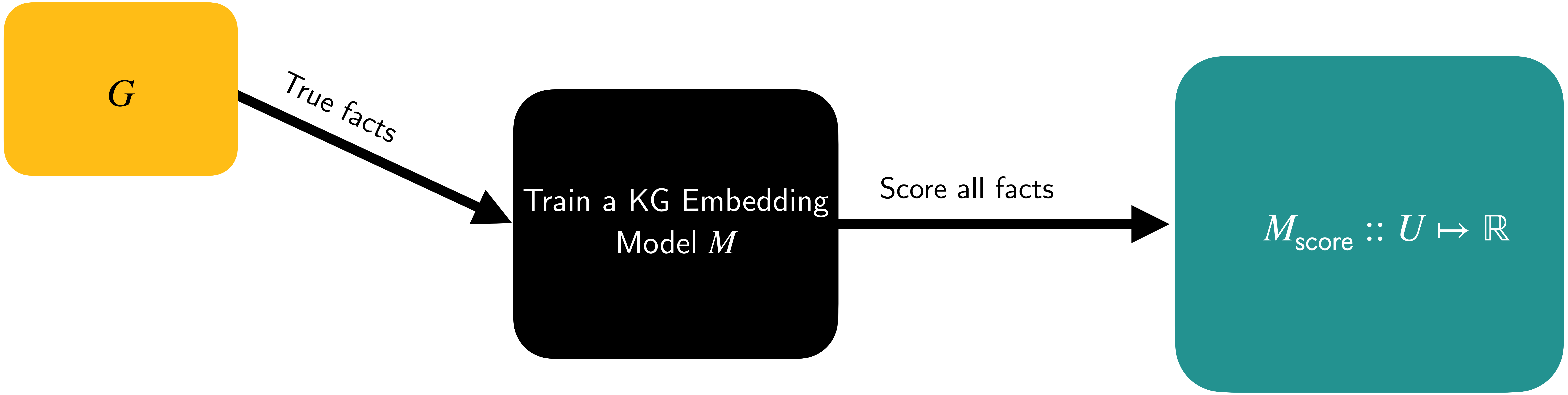
(iii) **Loss function**: What is the objective function to be minimised?

Well-known families of models classified in terms of model representation:

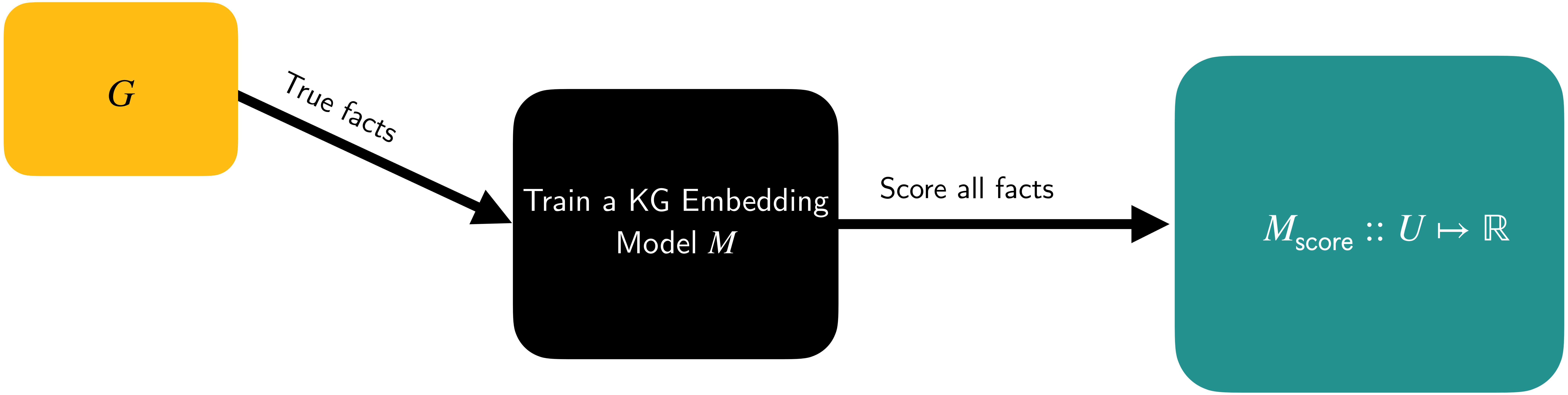
- **Translational models**: Embed entities as points in vector space, and model relations as translations operating on the embeddings of the entities.
- **Bilinear models**: Embed entities and relations into vector space, and model relations as a bilinear product between entity and relation embeddings.
- **Neural models**: Embed the entities and relations using a neural network (e.g., convolutional neural network).

KG Embedding Models: Basic Idea

KG Embedding Models: Basic Idea

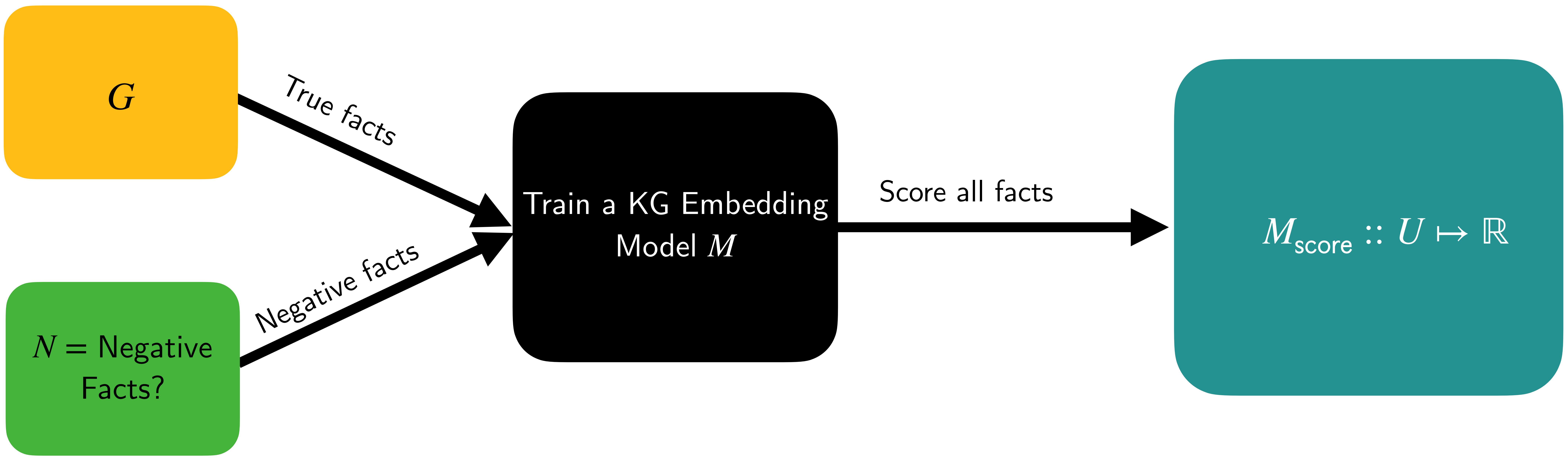


KG Embedding Models: Basic Idea



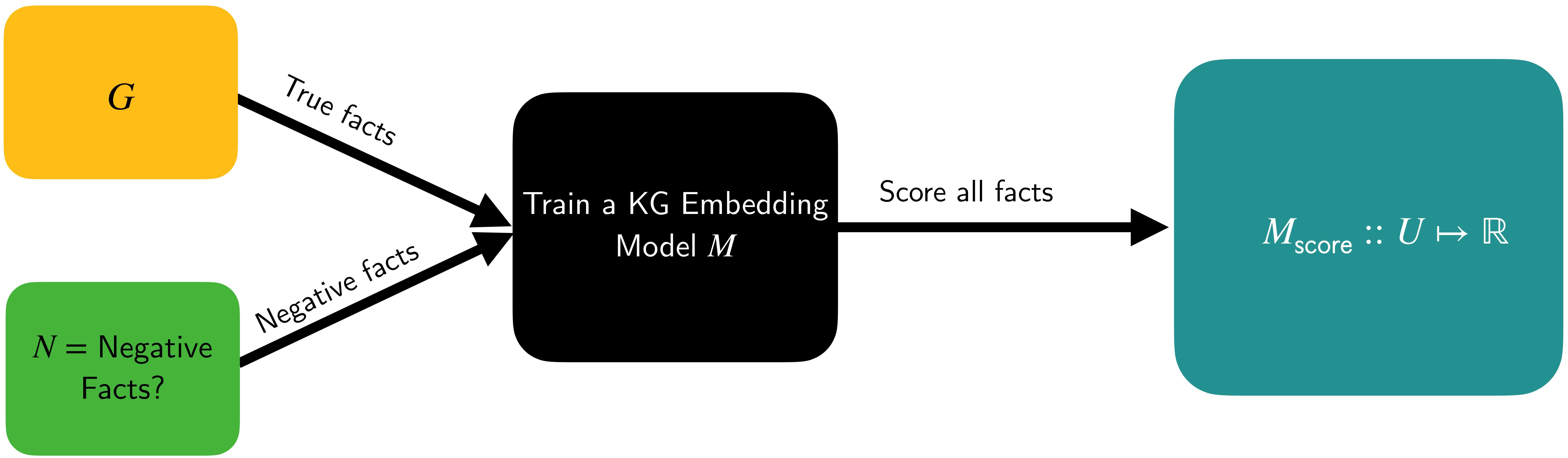
The main optimisation goal is find a vector configuration for entities and relationships so as to score/rank/evaluate "true facts" higher than "false facts" in accordance to a dissimilarity measure.

KG Embedding Models: Basic Idea



The main optimisation goal is find a vector configuration for entities and relationships so as to score/rank/evaluate “true facts” higher than “false facts” in accordance to a dissimilarity measure.

KG Embedding Models: Basic Idea



The main optimisation goal is find a vector configuration for entities and relationships so as to score/rank/evaluate "true facts" higher than "false facts" in accordance to a dissimilarity measure.

Problem: KGs typically store only positive information, and so encode only the facts that are true. There are no real negative examples to train with!

Negative Sampling

Negative Sampling

The idea is to corrupt true facts, and then use some of these corrupted facts as negative examples.

Negative Sampling

The idea is to corrupt true facts, and then use some of these corrupted facts as negative examples.

Given a KG G over entities E and relations R, every fact in G is called a true fact.

Negative Sampling

The idea is to corrupt true facts, and then use some of these corrupted facts as negative examples.

Given a KG G over entities E and relations R, every fact in G is called a true fact.

A corrupted fact is obtained by replacing only the head (resp., only the tail) entity in a true fact in G with an entity in E. Formally, for a true fact $r(h, t) \in G$, we define the set of all corrupted facts as:

Negative Sampling

The idea is to corrupt true facts, and then use some of these corrupted facts as negative examples.

Given a KG G over entities E and relations R, every fact in G is called a true fact.

A corrupted fact is obtained by replacing only the head (resp., only the tail) entity in a true fact in G with an entity in E. Formally, for a true fact $r(h, t) \in G$, we define the set of all corrupted facts as:

$$C^{r(h,t)} = \{r(e,t) \mid e \neq h \in \mathbf{E}, r(e,t) \notin G\} \cup \{r(h,e) \mid e \neq t \in \mathbf{E}, r(h,e) \notin G\}\,.$$

Negative Sampling

The idea is to corrupt true facts, and then use some of these corrupted facts as negative examples.

Given a KG G over entities E and relations R, every fact in G is called a true fact.

A corrupted fact is obtained by replacing only the head (resp., only the tail) entity in a true fact in G with an entity in E. Formally, for a true fact $r(h,t) \in G$, we define the set of all corrupted facts as:

$$C^{r(h,t)} = \{r(e,t) \mid e \neq h \in \mathbf{E}, r(e,t) \notin G\} \cup \{r(h,e) \mid e \neq t \in \mathbf{E}, r(h,e) \notin G\}.$$

A negative fact for a given true fact $r(h,t)$, is a fact randomly sampled from $C^{r(h,t)}$. The set of negative facts sampled for a given true fact $r(h,t)$ is denoted as $N^{r(h,t)}$.

Negative Sampling

The idea is to corrupt true facts, and then use some of these corrupted facts as negative examples.

Given a KG G over entities E and relations R, every fact in G is called a true fact.

A corrupted fact is obtained by replacing only the head (resp., only the tail) entity in a true fact in G with an entity in E. Formally, for a true fact $r(h,t) \in G$, we define the set of all corrupted facts as:

$$C^{r(h,t)} = \{r(e,t) \mid e \neq h \in \mathbf{E}, r(e,t) \notin G\} \cup \{r(h,e) \mid e \neq t \in \mathbf{E}, r(h,e) \notin G\}.$$

A negative fact for a given true fact $r(h,t)$, is a fact randomly sampled from $C^{r(h,t)}$. The set of negative facts sampled for a given true fact $r(h,t)$ is denoted as $N^{r(h,t)}$.

Sampling from corrupted facts and using these as negative facts is standard in the literature, and various sampling techniques are used, e.g., uniform sampling, adversarial sampling, etc.

Negative Sampling

The idea is to corrupt true facts, and then use some of these corrupted facts as negative examples.

Given a KG G over entities E and relations R, every fact in G is called a true fact.

A corrupted fact is obtained by replacing only the head (resp., only the tail) entity in a true fact in G with an entity in E. Formally, for a true fact $r(h,t) \in G$, we define the set of all corrupted facts as:

$$C^{r(h,t)} = \{r(e,t) \mid e \neq h \in \mathbf{E}, r(e,t) \notin G\} \cup \{r(h,e) \mid e \neq t \in \mathbf{E}, r(h,e) \notin G\} .$$

A negative fact for a given true fact $r(h,t)$, is a fact randomly sampled from $C^{r(h,t)}$. The set of negative facts sampled for a given true fact $r(h,t)$ is denoted as $N^{r(h,t)}$.

Sampling from corrupted facts and using these as negative facts is standard in the literature, and various sampling techniques are used, e.g., uniform sampling, adversarial sampling, etc.

Negative sampling is not ideal, as random sampling can clearly give a potentially correct fact as a negative fact, and require it to be ranked lower, misleadingly.

Model Expressiveness

Model Expressiveness

Model Expressiveness

A KGC model M is fully expressive if, for any given disjoint sets of true and false facts over a vocabulary (i.e., the ground truth of a set of facts), there exists a parameter configuration for M such that M accurately classifies all the given facts.

Model Expressiveness

A KGC model M is fully expressive if, for any given disjoint sets of true and false facts over a vocabulary (i.e., the ground truth of a set of facts), there exists a parameter configuration for M such that M accurately classifies all the given facts.

Intuitively, a fully expressive model can capture any ground truth of a given set of facts. Conversely, a model that is not fully expressive can fail to fit its training set properly, and thus can underfit.

Model Expressiveness

A KGC model M is fully expressive if, for any given disjoint sets of true and false facts over a vocabulary (i.e., the ground truth of a set of facts), there exists a parameter configuration for M such that M accurately classifies all the given facts.

Intuitively, a fully expressive model can capture any ground truth of a given set of facts. Conversely, a model that is not fully expressive can fail to fit its training set properly, and thus can underfit.

It is therefore desirable to have full expressivity, as we would our model to fit the dataset reasonably well, regardless of the complexity of the dataset.

Model Expressiveness

A KGC model M is fully expressive if, for any given disjoint sets of true and false facts over a vocabulary (i.e., the ground truth of a set of facts), there exists a parameter configuration for M such that M accurately classifies all the given facts.

Intuitively, a fully expressive model can capture any ground truth of a given set of facts. Conversely, a model that is not fully expressive can fail to fit its training set properly, and thus can underfit.

It is therefore desirable to have full expressivity, as we would our model to fit the dataset reasonably well, regardless of the complexity of the dataset.

Would theoretical inexpressivity surface in practice?

Model Expressiveness

A KGC model M is fully expressive if, for any given disjoint sets of true and false facts over a vocabulary (i.e., the ground truth of a set of facts), there exists a parameter configuration for M such that M accurately classifies all the given facts.

Intuitively, a fully expressive model can capture any ground truth of a given set of facts. Conversely, a model that is not fully expressive can fail to fit its training set properly, and thus can underfit.

It is therefore desirable to have full expressivity, as we would our model to fit the dataset reasonably well, regardless of the complexity of the dataset.

Would theoretical inexpressivity surface in practice?

Theoretical inexpressivity of a model may not surface empirically, especially if the benchmark datasets are not very complex. Knowing the expressive limitations of a model, however, it is easy to design datasets to empirically observe its limitations.

Model Inductive Capacity

Model Inductive Capacity

Model Inductive Capacity

Model inductive capacity is the generalisation capacity of a model, i.e., the quality of the predictions of the model over incomplete datasets.

Model Inductive Capacity

Model inductive capacity is the generalisation capacity of a model, i.e., the quality of the predictions of the model over incomplete datasets.

Full expressiveness does not necessarily correlate with inductive capacity: Fully expressive models can merely memorise training data and generalise poorly. It is important to develop models that are jointly fully expressive and have a strong inductive capacity.

Model Inductive Capacity

Model inductive capacity is the generalisation capacity of a model, i.e., the quality of the predictions of the model over incomplete datasets.

Full expressiveness does not necessarily correlate with inductive capacity: Fully expressive models can merely memorise training data and generalise poorly. It is important to develop models that are jointly fully expressive and have a strong inductive capacity.

How can model inductive capacity be studied?

Model Inductive Capacity

Model inductive capacity is the generalisation capacity of a model, i.e., the quality of the predictions of the model over incomplete datasets.

Full expressiveness does not necessarily correlate with inductive capacity: Fully expressive models can merely memorise training data and generalise poorly. It is important to develop models that are jointly fully expressive and have a strong inductive capacity.

How can model inductive capacity be studied?

Inference patterns are specifications of logical properties that may exist in a KG, which, if learned, enable further principled inferences from existing KG facts. Inference patterns are a common means to formally analyse the generalisation ability of KGC systems.

Model Inductive Capacity

Model inductive capacity is the generalisation capacity of a model, i.e., the quality of the predictions of the model over incomplete datasets.

Full expressiveness does not necessarily correlate with inductive capacity: Fully expressive models can merely memorise training data and generalise poorly. It is important to develop models that are jointly fully expressive and have a strong inductive capacity.

How can model inductive capacity be studied?

Inference patterns are specifications of logical properties that may exist in a KG, which, if learned, enable further principled inferences from existing KG facts. Inference patterns are a common means to formally analyse the generalisation ability of KGC systems.

One well-known example inference pattern is symmetry: A relation $r \in R$ is symmetric if, for any choice of entities $e_1, e_2 \in E$, whenever a fact $r(e_1, e_2)$ holds, then so does $r(e_2, e_1)$.

Model Inductive Capacity

Model inductive capacity is the generalisation capacity of a model, i.e., the quality of the predictions of the model over incomplete datasets.

Full expressiveness does not necessarily correlate with inductive capacity: Fully expressive models can merely memorise training data and generalise poorly. It is important to develop models that are jointly fully expressive and have a strong inductive capacity.

How can model inductive capacity be studied?

Inference patterns are specifications of logical properties that may exist in a KG, which, if learned, enable further principled inferences from existing KG facts. Inference patterns are a common means to formally analyse the generalisation ability of KGC systems.

One well-known example inference pattern is symmetry: A relation $r \in R$ is symmetric if, for any choice of entities $e_1, e_2 \in E$, whenever a fact $r(e_1, e_2)$ holds, then so does $r(e_2, e_1)$.

As a result, if a model learns a symmetry pattern for a relation r, then it can infer facts in the symmetric closure of r, thus providing a strong inductive bias.

Inference Patterns

Inference Patterns

An inference pattern specifies a logical property over a KG, which means that such patterns can be formalised using logical rules. To formalise this, let us extend our relational vocabulary over E and R with a set V of variables. A first-order atom is an expression of the form $r(x_i, x_j)$, where $r \in R$, and $x_i, x_j \in V$.

Inference Patterns

An inference pattern specifies a logical property over a KG, which means that such patterns can be formalised using logical rules. To formalise this, let us extend our relational vocabulary over E and R with a set V of variables. A first-order atom is an expression of the form $r(x_i, x_j)$, where $r \in R$, and $x_i, x_j \in V$.

A Boolean combination of first-order atoms is defined inductively using logical constructors $\neg, \wedge, \vee$, e.g., $\phi_1(x_1, x_3) = r_1(x_1, x_2) \wedge r_2(x_2, x_2)$ and $\phi_2(x_3, x_4) = r_2(x_3, x_4) \vee \neg r_3(x_4, x_3)$ are Boolean combinations of first-order atoms.

Inference Patterns

An inference pattern specifies a logical property over a KG, which means that such patterns can be formalised using logical rules. To formalise this, let us extend our relational vocabulary over E and R with a set V of variables. A first-order atom is an expression of the form $r(x_i, x_j)$, where $r \in R$, and $x_i, x_j \in V$.

A Boolean combination of first-order atoms is defined inductively using logical constructors $\neg, \wedge, \vee$, e.g., $\phi_1(x_1, x_3) = r_1(x_1, x_2) \wedge r_2(x_2, x_2)$ and $\phi_2(x_3, x_4) = r_2(x_3, x_4) \vee \neg r_3(x_4, x_3)$ are Boolean combinations of first-order atoms.

For the purposes of this lecture, we are interested in universally quantified first-order rules of the form:

$$\forall x_1 \ldots x_k \; \phi(x_1 \ldots x_k) \Rightarrow \psi(x_1 \ldots x_l),$$

with $k \geq l$. The semantics of such universally quantified first-order rules is that of first-order logic, restricted to a finite domain (as the set E of entities is finite).

Inference Patterns

Inference Patterns

We can express the symmetry inference pattern for a relation $r \in R$, in the form of such a logical rule as follows:

$$\forall x, y \; r(x, y) \Rightarrow r(y, x),$$

which holds if and only if the relation r is symmetric, i.e., the rule is invalidated if there exists two entities $e_1, e_2 \in E$, where $r(e_1, e_2)$ is true, but $r(e_2, e_1)$ is not.

Inference Patterns

We can express the symmetry inference pattern for a relation $r \in R$, in the form of such a logical rule as follows:

$$\forall x, y \; r(x, y) \Rightarrow r(y, x),$$

which holds if and only if the relation r is symmetric, i.e., the rule is invalidated if there exists two entities $e_1, e_2 \in E$, where $r(e_1, e_2)$ is true, but $r(e_2, e_1)$ is not.

Similarly, we can express that the relations $r_1, r_2 \in R$ are the inverse of each other in terms of two rules:

$$\forall x, y \; r_1(x, y) \Rightarrow r_2(y, x),$$

$$\forall x, y \; r_2(x, y) \Rightarrow r_1(y, x).$$

In this case, we will use the standard abbreviation $\Leftrightarrow$ and write $\forall x, y \; r_1(x, y) \Leftrightarrow r_2(y, x)$.

Inference Patterns

Inference pattern	**Inference rule**
Symmetry	$\forall x, y \ r(x, y) \Rightarrow r(y, x)$
Anti-symmetry	$\forall x, y \ r(x, y) \Rightarrow \neg r(y, x)$
Inversion	$\forall x, y \ r_1(x, y) \Leftrightarrow r_2(y, x)$
Composition	$\forall x, y, z \ r_1(x, y) \wedge r_2(y, z) \Rightarrow r_3(x, z)$
Hierarchy	$\forall x, y \ r_1(x, y) \Rightarrow r_2(x, y)$
Intersection	$\forall x, y \ r_1(x, y) \wedge r_2(x, y) \Rightarrow r_3(x, y)$
Mutual exclusion	$\forall x, y \ r_1(x, y) \Rightarrow \neg r_2(x, y)$

List of inference patterns commonly used in the literature and the corresponding logical rules. It is assumed that $r_1 \neq r_2 \neq r_3$.

Inference Patterns

Inference pattern	**Inference rule**
Symmetry	$\forall x, y \; r(x, y) \Rightarrow r(y, x)$
Anti-symmetry	$\forall x, y \; r(x, y) \Rightarrow \neg r(y, x)$
Inversion	$\forall x, y \; r_1(x, y) \Leftrightarrow r_2(y, x)$
Composition	$\forall x, y, z \; r_1(x, y) \wedge r_2(y, z) \Rightarrow r_3(x, z)$
Hierarchy	$\forall x, y \; r_1(x, y) \Rightarrow r_2(x, y)$
Intersection	$\forall x, y \; r_1(x, y) \wedge r_2(x, y) \Rightarrow r_3(x, y)$
Mutual exclusion	$\forall x, y \; r_1(x, y) \Rightarrow \neg r_2(x, y)$

List of inference patterns commonly used in the literature and the corresponding logical rules. It is assumed that $r_1 \neq r_2 \neq r_3$.

These patterns are very prominent in many datasets. While these patterns and the corresponding rules are not very expressive, they already are a challenge for KGE models, as it is already hard for existing systems to capture these patterns.

Empirical Evaluation

Empirical Evaluation: Ranking

Empirical Evaluation: Ranking

The most common empirical evaluation task for KGE methods is based on entity *ranking*. The knowledge graph G is partitioned into a set of training (G_{tr}), validation (G_v), and test facts (G_{test}).

Empirical Evaluation: Ranking

The most common empirical evaluation task for KGE methods is based on entity *ranking*. The knowledge graph G is partitioned into a set of training (G_{tr}), validation (G_v), and test facts (G_{test}).

Given a test fact $r(h, t) \in G_{test}$, we consider the following sets:

Empirical Evaluation: Ranking

The most common empirical evaluation task for KGE methods is based on entity *ranking*. The knowledge graph G is partitioned into a set of training (G_{tr}), validation (G_v), and test facts (G_{test}).

Given a test fact $r(h, t) \in G_{test}$, we consider the following sets:

$$\begin{aligned} r(\_, t) = \; & \{r(e, t) \mid e \in \mathbf{E}, r(e, t) \notin G_{tr} \cup G_v \cup G_{test}\} \cup \{r(h, t)\}, \\ r(h, \_) = \; & \{r(h, e) \mid e \in \mathbf{E}, r(h, e) \notin G_{tr} \cup G_v \cup G_{test}\} \cup \{r(h, t)\} . \end{aligned}$$

Empirical Evaluation: Ranking

The most common empirical evaluation task for KGE methods is based on entity *ranking*. The knowledge graph G is partitioned into a set of training (G_{tr}), validation (G_v), and test facts (G_{test}).

Given a test fact $r(h,t) \in G_{test}$, we consider the following sets:

$$
\begin{aligned}
r(\_,t) = \; & \{r(e,t) \mid e \in \mathbf{E}, r(e,t) \notin G_{tr} \cup G_v \cup G_{test}\} \cup \{r(h,t)\}, \\
r(h,\_) = \; & \{r(h,e) \mid e \in \mathbf{E}, r(h,e) \notin G_{tr} \cup G_v \cup G_{test}\} \cup \{r(h,t)\}\,.
\end{aligned}
$$

Importantly, all facts that occur in the training, validation, or test data are filtered out from these sets (except the test fact itself). This is to ensure that other facts known to be true do not affect the ranking. This is the so-called filtered evaluation which has become standard practice in experimental evaluation (Bordes et al., 2013).

Empirical Evaluation: Ranking

The most common empirical evaluation task for KGE methods is based on entity *ranking*. The knowledge graph G is partitioned into a set of training (G_{tr}), validation (G_v), and test facts (G_{test}).

Given a test fact $r(h,t) \in G_{test}$, we consider the following sets:

$$\begin{aligned} r(\_,t) = &\ \{r(e,t) \mid e \in \mathbf{E}, r(e,t) \notin G_{tr} \cup G_v \cup G_{test}\} \cup \{r(h,t)\}, \\ r(h,\_) = &\ \{r(h,e) \mid e \in \mathbf{E}, r(h,e) \notin G_{tr} \cup G_v \cup G_{test}\} \cup \{r(h,t)\}\,. \end{aligned}$$

Importantly, all facts that occur in the training, validation, or test data are filtered out from these sets (except the test fact itself). This is to ensure that other facts known to be true do not affect the ranking. This is the so-called filtered evaluation which has become standard practice in experimental evaluation (Bordes et al., 2013).

Every fact in these sets is ranked in accordance to the scoring function of the model in descending order.

Empirical Evaluation: Ranking

The most common empirical evaluation task for KGE methods is based on entity *ranking*. The knowledge graph G is partitioned into a set of training (G_{tr}), validation (G_v), and test facts (G_{test}).

Given a test fact $r(h,t) \in G_{test}$, we consider the following sets:

$$
\begin{aligned}
r(\_,t) = & \ \{r(e,t) \mid e \in \mathbf{E}, r(e,t) \notin G_{tr} \cup G_v \cup G_{test}\} \cup \{r(h,t)\}, \\
r(h,\_) = & \ \{r(h,e) \mid e \in \mathbf{E}, r(h,e) \notin G_{tr} \cup G_v \cup G_{test}\} \cup \{r(h,t)\} .
\end{aligned}
$$

Importantly, all facts that occur in the training, validation, or test data are filtered out from these sets (except the test fact itself). This is to ensure that other facts known to be true do not affect the ranking. This is the so-called filtered evaluation which has become standard practice in experimental evaluation (Bordes et al., 2013).

Every fact in these sets is ranked in accordance to the scoring function of the model in descending order.

The rank of the entity e relative to the facts $r(\_,t)$, denoted $rank(e \mid r(\_,t))$, is the rank of the fact $r(e,t)$ in $r(\_,t)$; similarly, the rank of the entity e relative to the facts $r(h,\_)$, denoted $rank(e \mid r(h,\_))$, is the rank of the fact $r(h,e)$ in $r(h,\_)$.

Empirical Evaluation: Metrics

Empirical Evaluation: Metrics

Mean rank (MR) is the average rank of true facts against their corrupted counterparts:

$$\frac{1}{2 \mid G_{test} \mid} \sum_{r(h,t) \in G_{test}} \big(rank(h \mid r(\_, t)) + rank(t \mid r(h, \_)) \big)$$

Empirical Evaluation: Metrics

Mean rank (MR) is the average rank of true facts against their corrupted counterparts:

$$\frac{1}{2\,|\,G_{test}\,|}\sum_{r(h,t)\in G_{test}}\big(rank(h \mid r(\_,t)) + rank(t \mid r(h,\_))\big)$$

Mean reciprocal rank (MRR) is the inverse average rank of true facts against their corrupted counterparts:

$$\frac{1}{2\,|\,G_{test}\,|}\sum_{r(h,t)\in G_{test}}\left(\frac{1}{rank(h \mid r(\_,t))} + \frac{1}{rank(t \mid r(h,\_))}\right)$$

Empirical Evaluation: Metrics

Mean rank (MR) is the average rank of true facts against their corrupted counterparts:

$$\frac{1}{2 \mid G_{test} \mid} \sum_{r(h,t) \in G_{test}} \left(rank(h \mid r(\_, t)) + rank(t \mid r(h, \_)) \right)$$

Mean reciprocal rank (MRR) is the inverse average rank of true facts against their corrupted counterparts:

$$\frac{1}{2 \mid G_{test} \mid} \sum_{r(h,t) \in G_{test}} \left(\frac{1}{rank(h \mid r(\_, t))} + \frac{1}{rank(t \mid r(h, \_))} \right)$$

Hits@k is the proportion of true facts with rank at most k:

$$\frac{1}{2 \mid G_{test} \mid} \sum_{r(h,t) \in G_{test}} \left(\mathbf{1}(rank(h \mid r(\_, t)) \leq k) + \mathbf{1}(rank(t \mid r(h, \_)) \leq k) \right),$$

where $\mathbf{1}(c)$ is the indicator function that returns 1, if c is true, and 0, otherwise.

Empirical Evaluation: Datasets

Empirical Evaluation: Datasets

FB15k (Bordes et al., 2013): A subset of Freebase (Bollacker et al., 2008), where a large part of the test facts $r(x, y)$ can be directly inferred via an inverse relation $r'(y, x)$, which makes the inversion pattern very prominent (Toutanova & Chen, 2015). Other patterns on FB15k are symmetry/antisymmetry and composition patterns.

Empirical Evaluation: Datasets

FB15k (Bordes et al., 2013): A subset of Freebase (Bollacker et al., 2008), where a large part of the test facts $r(x, y)$ can be directly inferred via an inverse relation $r'(y, x)$, which makes the inversion pattern very prominent (Toutanova & Chen, 2015). Other patterns on FB15k are symmetry/antisymmetry and composition patterns.

FB15K-237 (Toutanova & Chen, 2015): A subset of FB15k , where inverse relations are deleted. The prominent patterns are composition and symmetry/antisymmetry patterns.

Empirical Evaluation: Datasets

FB15k (Bordes et al., 2013): A subset of Freebase (Bollacker et al., 2008), where a large part of the test facts $r(x, y)$ can be directly inferred via an inverse relation $r'(y, x)$, which makes the inversion pattern very prominent (Toutanova & Chen, 2015). Other patterns on FB15k are symmetry/antisymmetry and composition patterns.

FB15K-237 (Toutanova & Chen, 2015): A subset of FB15k , where inverse relations are deleted. The prominent patterns are composition and symmetry/antisymmetry patterns.

WN18 (Bordes et al., 2013): A subset of WordNet (Miller, 1995), featuring lexical relations between words. This dataset has also many inverse relations, and the main inference patterns are symmetry/antisymmetry and inversion.

Empirical Evaluation: Datasets

FB15k (Bordes et al., 2013): A subset of Freebase (Bollacker et al., 2008), where a large part of the test facts $r(x, y)$ can be directly inferred via an inverse relation $r'(y, x)$, which makes the inversion pattern very prominent (Toutanova & Chen, 2015). Other patterns on FB15k are symmetry/antisymmetry and composition patterns.

FB15K-237 (Toutanova & Chen, 2015): A subset of FB15k , where inverse relations are deleted. The prominent patterns are composition and symmetry/antisymmetry patterns.

WN18 (Bordes et al., 2013): A subset of WordNet (Miller, 1995), featuring lexical relations between words. This dataset has also many inverse relations, and the main inference patterns are symmetry/antisymmetry and inversion.

WN18RR (Dettmers et al., 2017): A subset of WN18, where inverse relations are deleted. The prominent inference patterns are symmetry/antisymmetry and composition.

Empirical Evaluation: Datasets

FB15k (Bordes et al., 2013): A subset of Freebase (Bollacker et al., 2008), where a large part of the test facts $r(x, y)$ can be directly inferred via an inverse relation $r'(y, x)$, which makes the inversion pattern very prominent (Toutanova & Chen, 2015). Other patterns on FB15k are symmetry/antisymmetry and composition patterns.

FB15K-237 (Toutanova & Chen, 2015): A subset of FB15k , where inverse relations are deleted. The prominent patterns are composition and symmetry/antisymmetry patterns.

WN18 (Bordes et al., 2013): A subset of WordNet (Miller, 1995), featuring lexical relations between words. This dataset has also many inverse relations, and the main inference patterns are symmetry/antisymmetry and inversion.

WN18RR (Dettmers et al., 2017): A subset of WN18, where inverse relations are deleted. The prominent inference patterns are symmetry/antisymmetry and composition.

YAGO3-10: A subset of the YAGO3 (Mahdisoltani et al., 2015), where all entities appear in at least 10 facts.

Empirical Evaluation: Datasets

Dataset	|E|	|R|	Training facts	Validation facts	Test facts
FB15K-237	14,541	237	272,115	17,535	20,466
WN18RR	40,943	11	86,835	3,034	3,034
YAGO3-10	123,182	37	1,079,040	5,000	5,000

Empirical Evaluation: Datasets

Dataset	|E|	|R|	Training facts	Validation facts	Test facts
FB15K-237	14,541	237	272,115	17,535	20,466
WN18RR	40,943	11	86,835	3,034	3,034
YAGO3-10	123,182	37	1,079,040	5,000	5,000

Datasets with their respective #entities ($|\mathbf{E}|$), #relations ($|\mathbf{R}|$), and #facts.

Summary

Summary

- Relational data is prominent in real-world applications!

Summary

- Relational data is prominent in real-world applications!
- Discussed KG embedding models through the lens of the KG completion task.

Summary

- Relational data is prominent in real-world applications!
- Discussed KG embedding models through the lens of the KG completion task.
- The families of translational, bilinear, and neural models are briefly discussed.

Summary

- Relational data is prominent in real-world applications!
- Discussed KG embedding models through the lens of the KG completion task.
- The families of translational, bilinear, and neural models are briefly discussed.
- Established evaluation criteria for different models:

Summary

- Relational data is prominent in real-world applications!
- Discussed KG embedding models through the lens of the KG completion task.
- The families of translational, bilinear, and neural models are briefly discussed.
- Established evaluation criteria for different models:
  - Model expressiveness

Summary

- Relational data is prominent in real-world applications!
- Discussed KG embedding models through the lens of the KG completion task.
- The families of translational, bilinear, and neural models are briefly discussed.
- Established evaluation criteria for different models:
  - Model expressiveness
  - Model inductive capacity and inference patterns

Summary

- Relational data is prominent in real-world applications!
- Discussed KG embedding models through the lens of the KG completion task.
- The families of translational, bilinear, and neural models are briefly discussed.
- Established evaluation criteria for different models:
    - Model expressiveness
    - Model inductive capacity and inference patterns
    - Empirical evaluation: Datasets and metrics

Summary

- Relational data is prominent in real-world applications!
- Discussed KG embedding models through the lens of the KG completion task.
- The families of translational, bilinear, and neural models are briefly discussed.
- Established evaluation criteria for different models:
  - Model expressiveness
  - Model inductive capacity and inference patterns
  - Empirical evaluation: Datasets and metrics
- We have not introduced/evaluated any specific model: Next lecture!

References

- A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. *NIPS*, 2013.
- K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. *MOD*, 2008.
- T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *NIPS*, 2013.
- G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 1995.
- T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel. Convolutional 2D knowledge graph embeddings. *AAAI*, 2018.
- K. Toutanova and D. Chen. Observed versus latent features for knowledge base and text inference. Proc. of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, 2015.
- F. Mahdisoltani, J. Biega, and F. Suchanek. YAGO3: A Knowledge Base from Multilingual Wikipedias. *CIDR,* 2015.