DSC250: Advanced Data Mining

Knowledge Graphs (KGs)

Zhiting Hu

Lecture 14, Feb 20, 2025

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

Outline

- Knowledge graphs

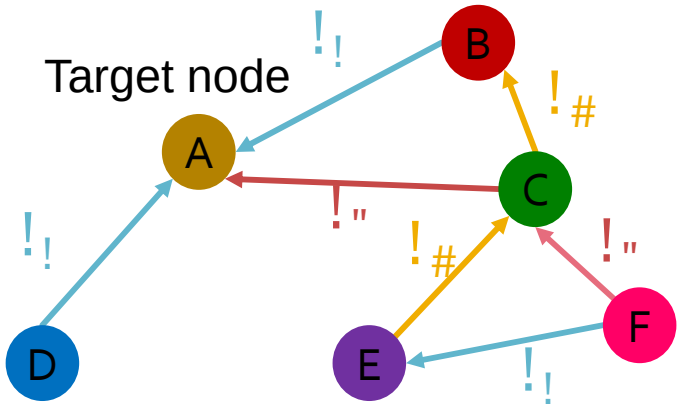
- Presentation
  - Lingyi Yang, Jiayue Yuan: "Generative Agents: Interactive Simulacra of Human Behavior"
  - Yuyuan Wu, Zijie Feng: "Data-efficient Fine-tuning for LLM-based Recommendation"
  - Ivy Nguyen, Joshua Chuang: "Prediction of COVID-19 cases by multifactor driven long short-term memory (LSTM) model"
  - Shuyu Wang, Caroline Zhang: "Yo'LLaVA: Your Personalized Language and Vision Assistant"

Outline

- Overview
- Knowledge Graph Completion (Link Prediction)
- Reasoning on Knowledge Graphs

Heterogeneous Graphs

- **Heterogeneous graphs: a graph with multiple relation types**



Input graph

Heterogeneous Graphs

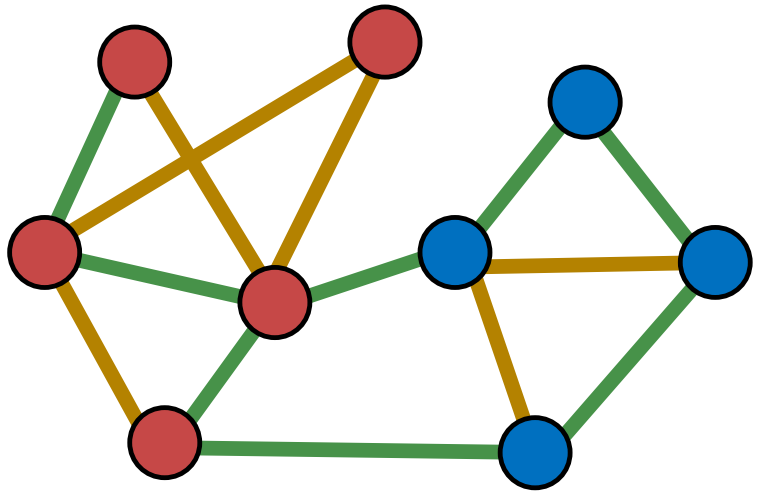
8 possible relation types!

(Paper, Cite, Paper)

(Paper, Like, Paper)

(Paper, Cite, Author)

(Paper, Like, Author)



(Author, Cite, Author)

(Author, Like, Author)

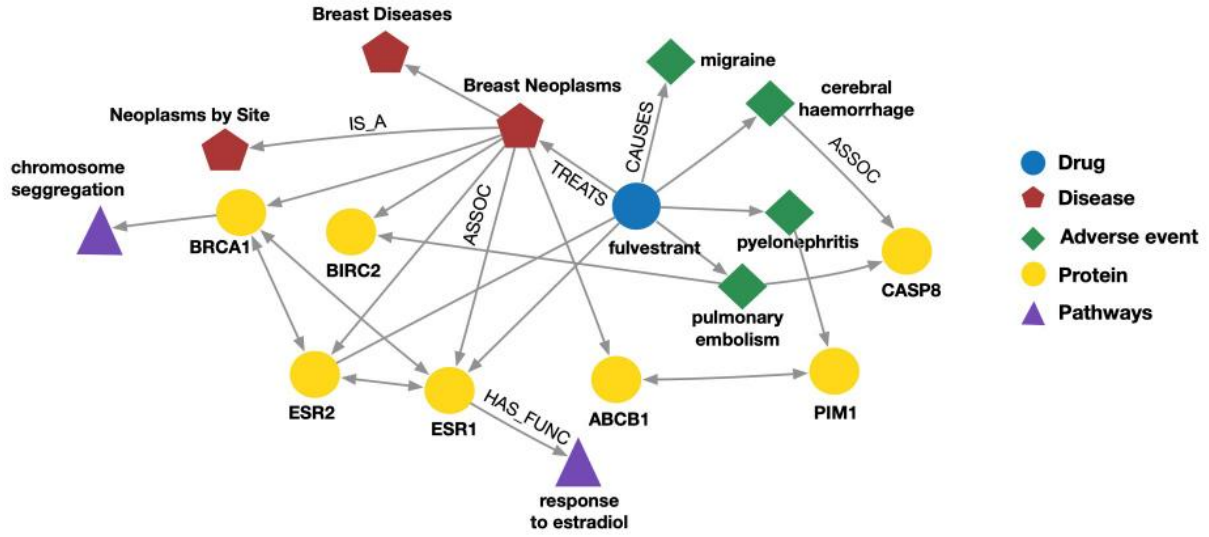
(Author, Cite, Paper)

(Author, Like, Paper)

Relation types: (node_start, edge, node_end)

- We use **relation type to describe an edge** (as opposed to edge type)
- Relation type better captures the interaction between nodes and edges

Heterogeneous Graphs



Biomedical Knowledge Graphs

Example node: Migraine

Example relation: (fulvestrant, Treats, Breast Neoplasms)

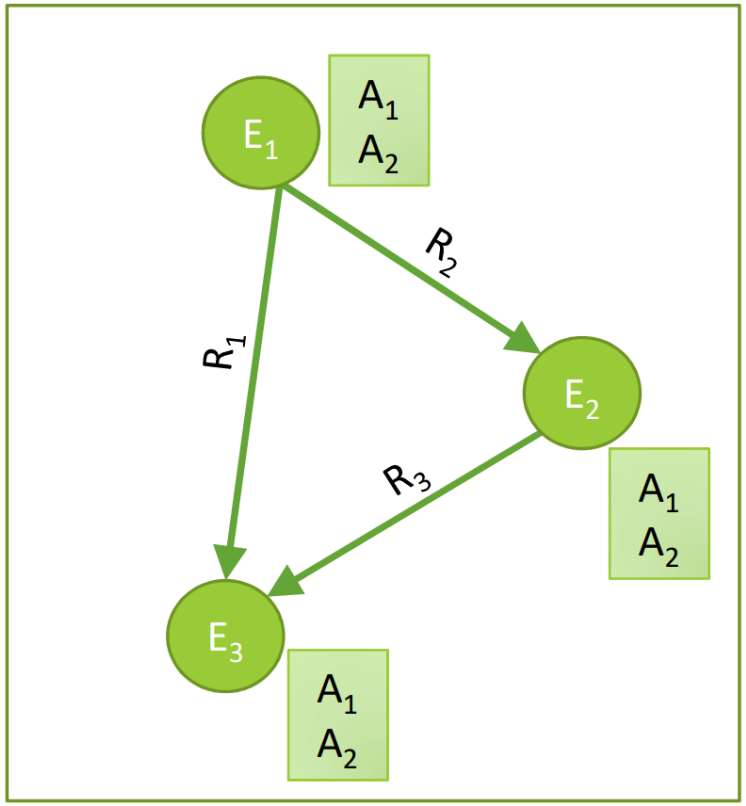
Example node type: Protein

Example edge type: Causes

Knowledge Graph

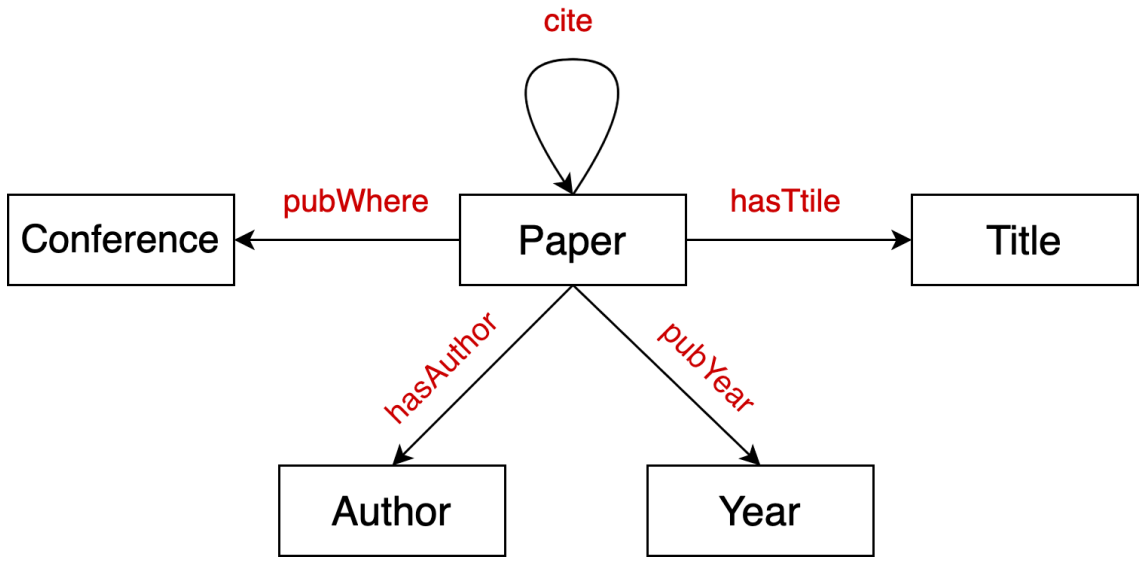
Knowledge in graph form:

- Capture entities, types, and relationships
- Nodes are **entities**
- Nodes are labeled with their **types**
- Edges between two nodes capture **relationships** between entities
- **KG is an example of a heterogeneous graph**



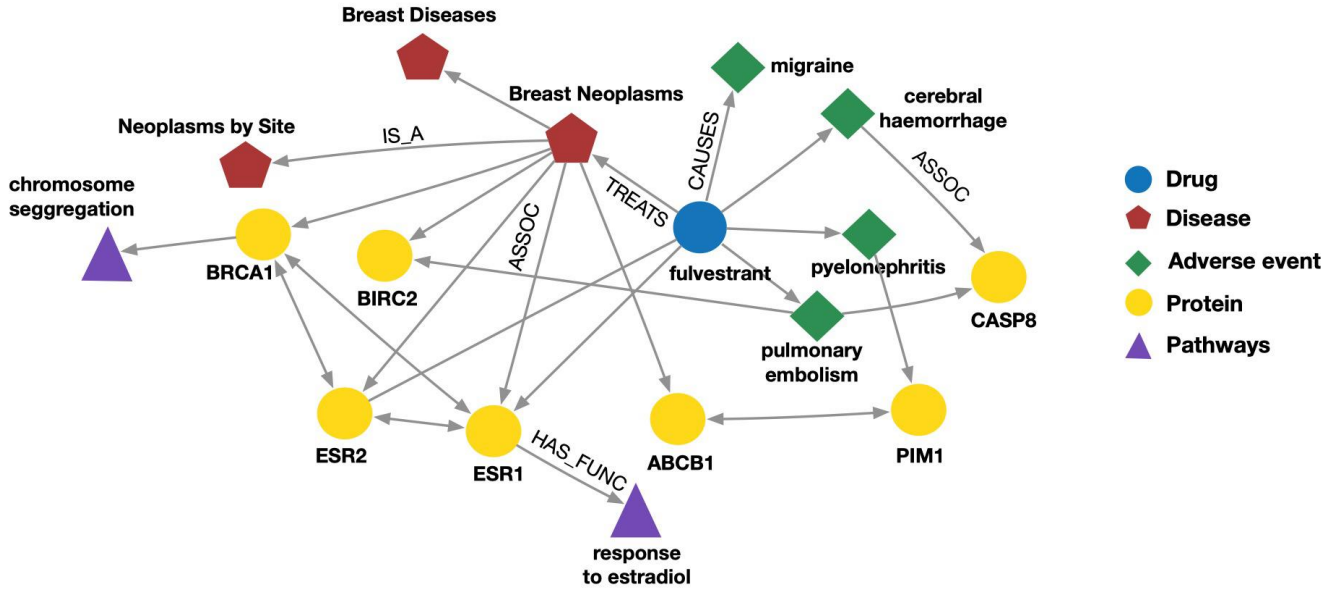
Example: Bibliographic Networks

- **Node types**: paper, title, author, conference, year
- **Relation types**: pubWhere, pubYear, hasTitle, hasAuthor, cite



Example: Bio Knowledge Graphs

- **Node types**: drug, disease, adverse event, protein, pathways
- **Relation types**: has_func, causes, assoc, treats, is_a



KGs in Practice

Examples of knowledge graphs

- Google Knowledge Graph
- Amazon Product Graph
- Facebook Graph API
- IBM Watson
- Microsoft Satori
- Project Hanover/Literome
- LinkedIn Knowledge Graph
- Yandex Object Answer

Applications of KGs

- **Serving information:**

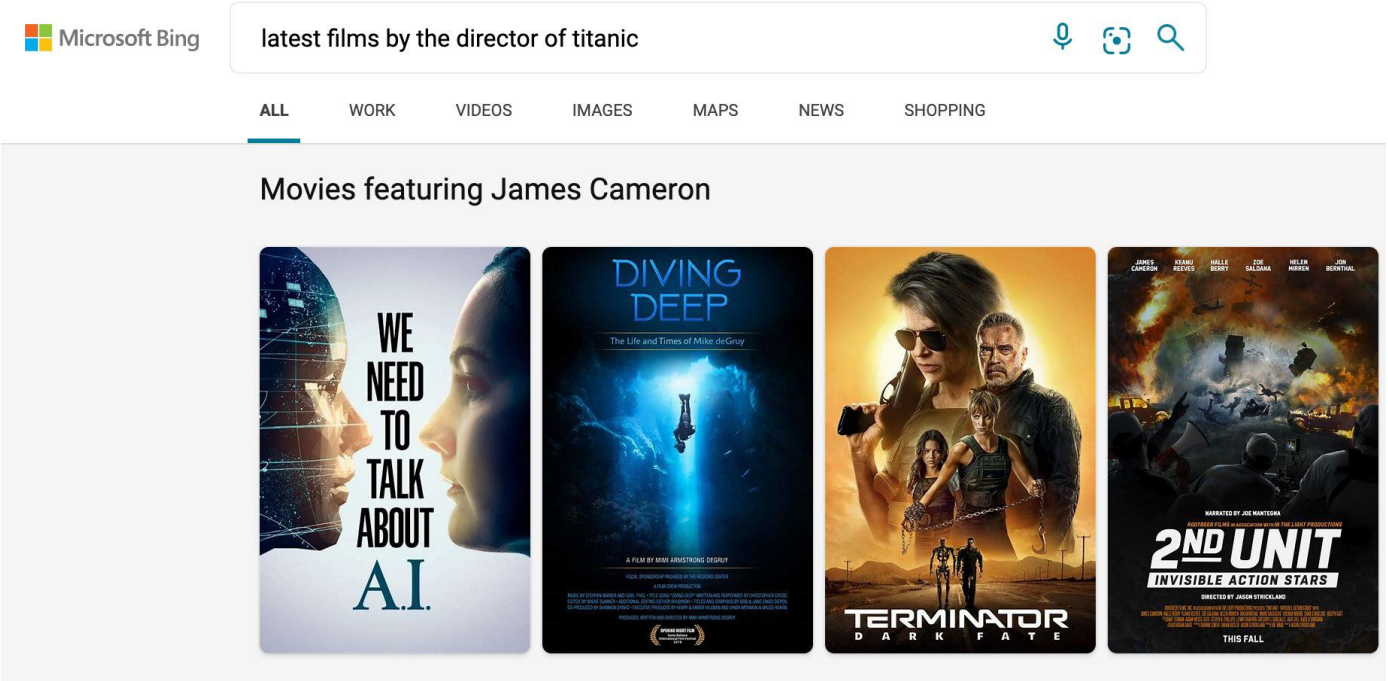


Image credit: Bing

Applications of KGs

- **Question answering and conversation agents**

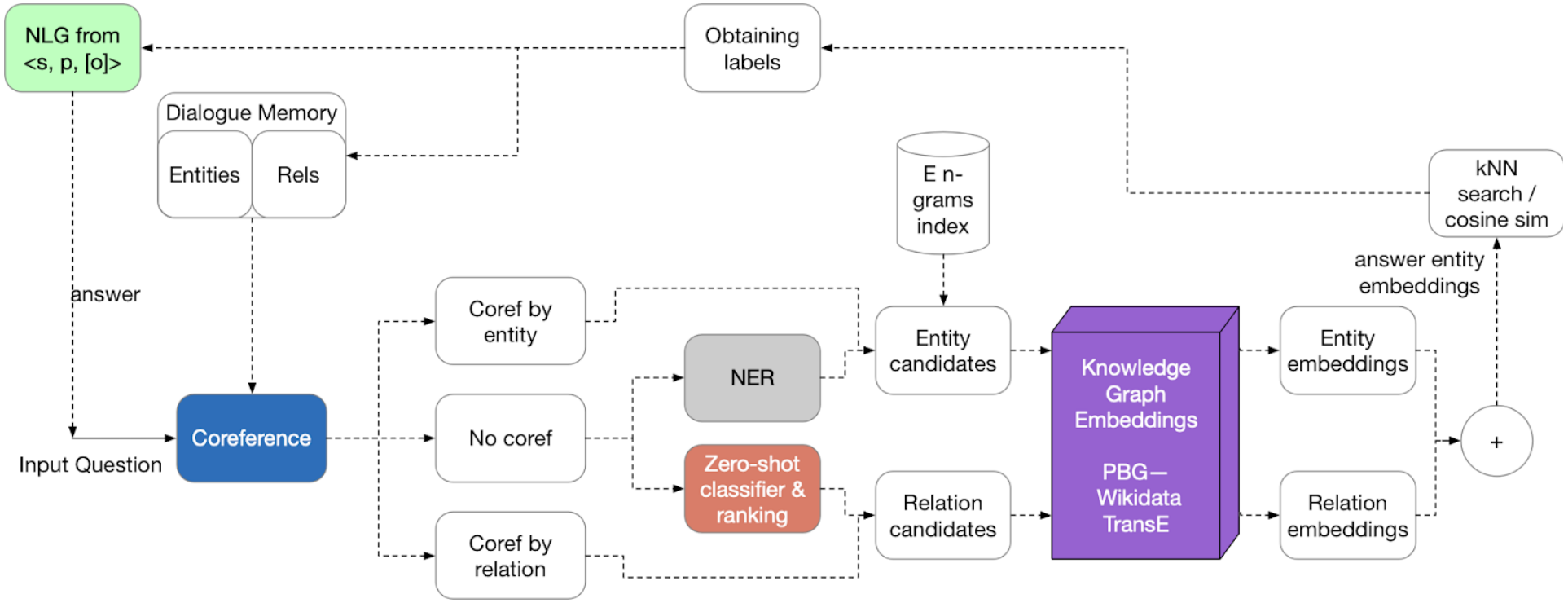


Image credit: Medium

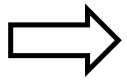
KG Datasets

- **Publicly available KGs:**
  - FreeBase, Wikidata, Dbpedia, YAGO, NELL, etc.
- **Common characteristics:**
  - **Massive**: Millions of nodes and edges
  - **Incomplete**: Many true edges are missing

KG Datasets

- **Publicly available KGs:**
  - FreeBase, Wikidata, Dbpedia, YAGO, NELL, etc.
- **Common characteristics:**
  - **Massive**: Millions of nodes and edges
  - **Incomplete**: Many true edges are missing

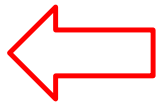
Given a massive KG, enumerating all the possible facts is intractable!



Can we predict plausible BUT missing links?

Example: Freebase

- **Freebase**
  - ~80 million **entities**
  - ~38K **relation types**
  - ~3 billion **facts/triples**



93.8% of persons from Freebase have no place of birth and 78.5% have no nationality!

- **Datasets:** FB15k/FB15k-237
  - A **complete** subset of Freebase, used by researchers to learn KG models

Dataset	Entities	Relations	Total Edges
FB15k	14,951	1,345	592,213
FB15k-237	14,505	237	310,079

[1] Paulheim, Heiko. "Knowledge graph refinement: A survey of approaches and evaluation methods." *Semantic web* 8.3 (2017): 489-508.
[2] Min, Bonan, et al. "Distant supervision for relation extraction with an incomplete knowledge base." *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2013.

Outline

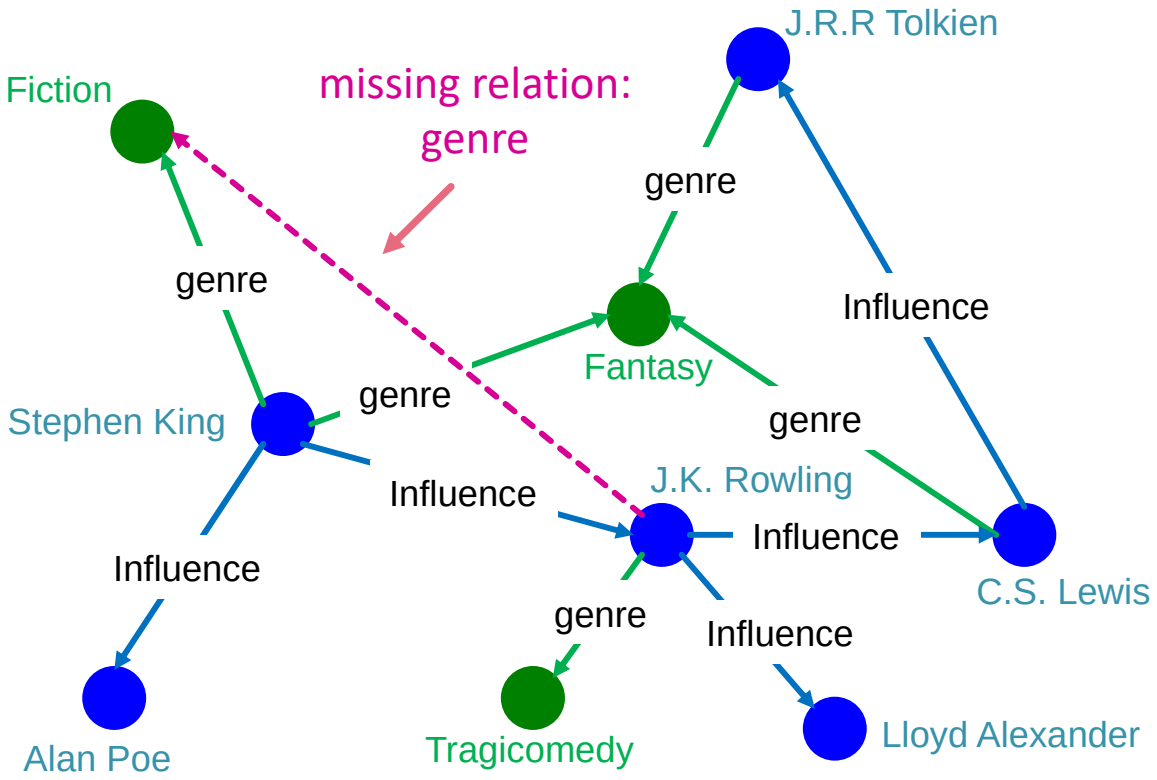
- Overview
- **Knowledge Graph Completion (Link Prediction)**
- Reasoning on Knowledge Graphs

KG Completion Task

Given an enormous KG, can we complete the KG?

- For a given (**head**, **relation**), we predict missing **tails**.
  - (Note this is slightly different from link prediction task)

Example task: predict the tail "Science Fiction" for ("J.K. Rowling", "genre")



KG Representation

- Edges in KG are represented as **triples** $(h, \$, \%)$
  - head (h) has relation $(\$)$ with tail $(\%)$

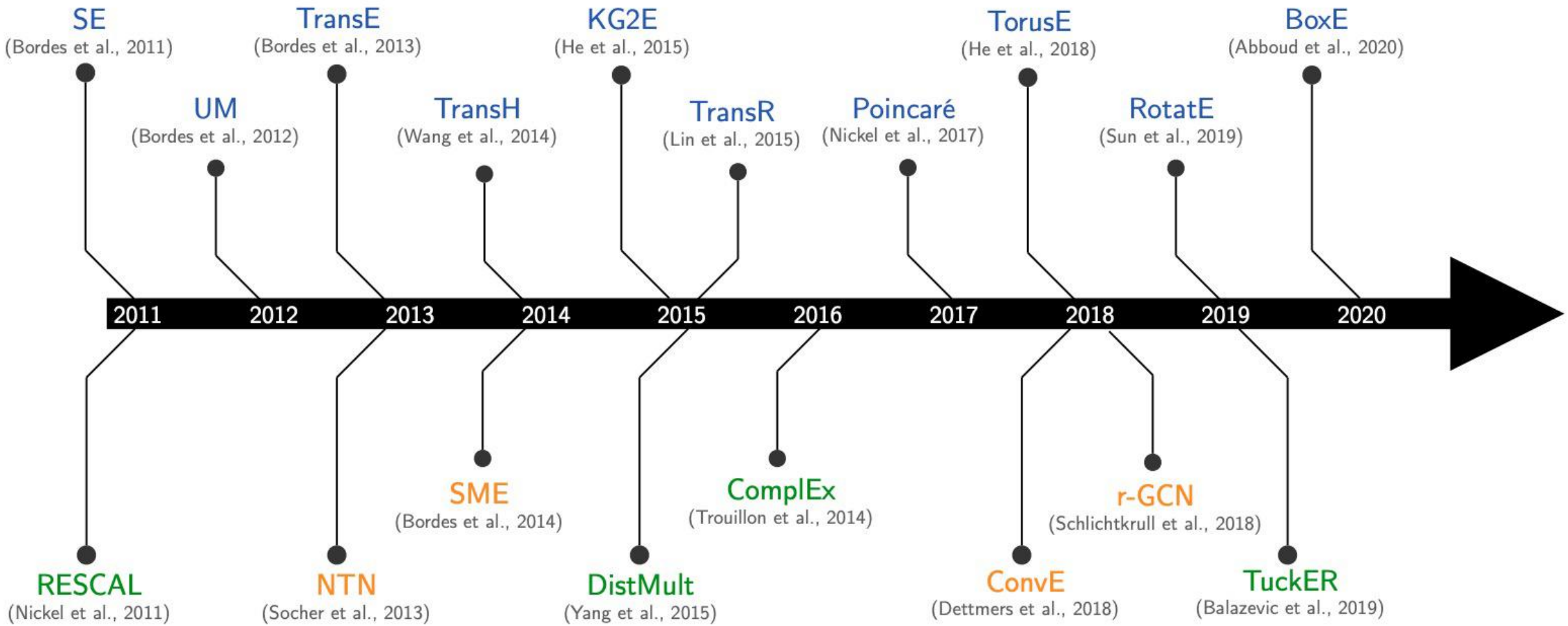
KG Representation

- Edges in KG are represented as **triples** $(h, \$, \%)$
  - head (h) has relation $(\$)$ with tail $(\%)$
- **Key Idea:**
  - Model entities and relations in embedding space $\mathbb{R}^{!}$
    - Associate entities and relations with **shallow embeddings**
      - **Note we do not learn a GNN here!**

KG Representation

- Edges in KG are represented as **triples** $(h, \$, \%)$
  - head (h) has relation $(\$)$ with tail $(\%)$
- **Key Idea:**
  - Model entities and relations in embedding space $\mathbb{R}^!$
    - Associate entities and relations with **shallow embeddings**
      - **Note we do not learn a GNN here!**
  - Given a triple $(h, \$, \%)$, the goal is that the embedding of $(h, \$)$ **should be close** to the embedding of $\%$
    - How to embed $(h, \#)$?
    - How to define score $\$_{\$}(h, \%)$?
      - Score $"_!$ is high if $(h, \&')$ exists, else $"_!$ is low

Many KG Embedding Methods



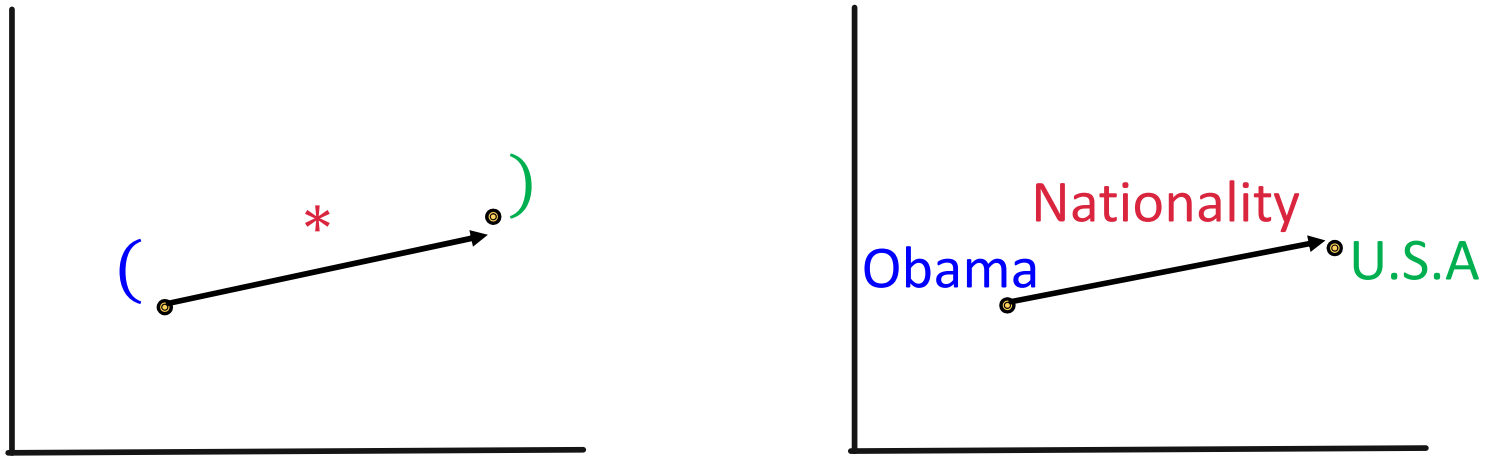
TransE for KG Completion

- **Intuition: Translation**

  For a triple (h, r, t), let $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^!$ be embedding vectors.

  embedding vectors will appear in boldface

- **TransE:** $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ if the given link exists else $\mathbf{h} + \mathbf{r} \neq \mathbf{t}$



Bordes et al., Translating embeddings for modeling multi-relational data, NeurIPS 2013.

TransE for KG Completion

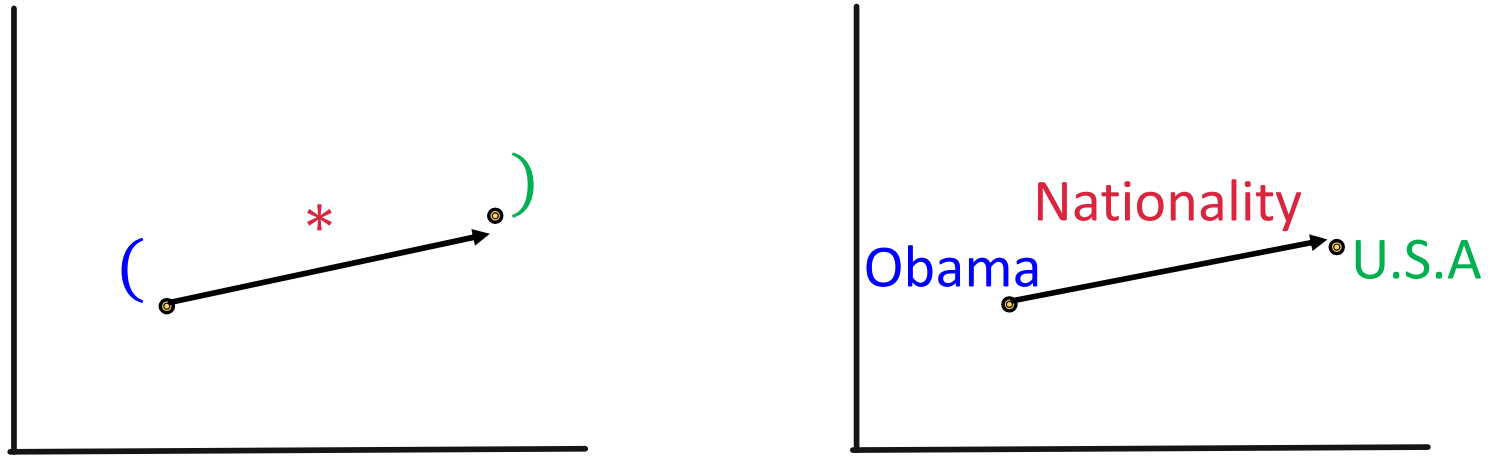
- **Intuition: Translation**

  For a triple (h, r, t), let $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^!$ be embedding vectors.

  embedding vectors will appear in boldface

- **TransE:** $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ if the given link exists else $\mathbf{h} + \mathbf{r} \neq \mathbf{t}$

Entity scoring function: $f_{"}(h, t) = -||\mathbf{h} + \mathbf{r} - \mathbf{t}||$



Bordes et al., Translating embeddings for modeling multi-relational data, NeurIPS 2013.

Connectivity Patterns in KG

- **Relations in a heterogeneous KG have different properties:**
  - Example:
    - **Symmetry:** If the edge $(h, \text{"Roommate"}, t)$ exists in KG, then the edge $(t, \text{"Roommate"}, h)$ should also exist.
    - **Inverse relation:** If the edge $(h, \text{"Advisor"}, t)$ exists in KG, then the edge $(t, \text{"Advisee"}, h)$ should also exist.

Connectivity Patterns in KG

- **Relations in a heterogeneous KG have different properties:**
  - Example:
    - **Symmetry:** If the edge $(h, \text{"Roommate"}, t)$ exists in KG, then the edge $(t, \text{"Roommate"}, h)$ should also exist.
    - **Inverse relation:** If the edge $(h, \text{"Advisor"}, t)$ exists in KG, then the edge $(t, \text{"Advisee"}, h)$ should also exist.
- Can we **categorize** these relation patterns?
- **Are KG embedding methods (e.g., TransE) expressive enough to model these patterns?**

Four Relationship Patterns

- **Symmetric (Antisymmetric) Relations:**

$$r(h,t) \Rightarrow r(t,h) \quad (r(h,t) \Rightarrow \neg r(t,h)) \quad \forall h,t$$

  - **Example**:
    - Symmetric: Family, Roommate
    - Antisymmetric: Hypernym (a word with a broader meaning: poodle vs. dog)
- **Inverse Relations:**

$$r_{(}(h,t) \Rightarrow r_{)}(t,h)$$

  - **Example** : (Advisor, Advisee)
- **Composition (Transitive) Relations:**

$$r_{)}(x,y) \wedge r_{(}(y,z) \Rightarrow r_{*}(x,z) \quad \forall x,y,z$$

  - **Example**: My mother's husband is my father.
- **1-to-N relations:**

$$r(h,t_{)}), r(h,t_{(}), \ldots, r(h,t_{+}) \text{ are all True.}$$

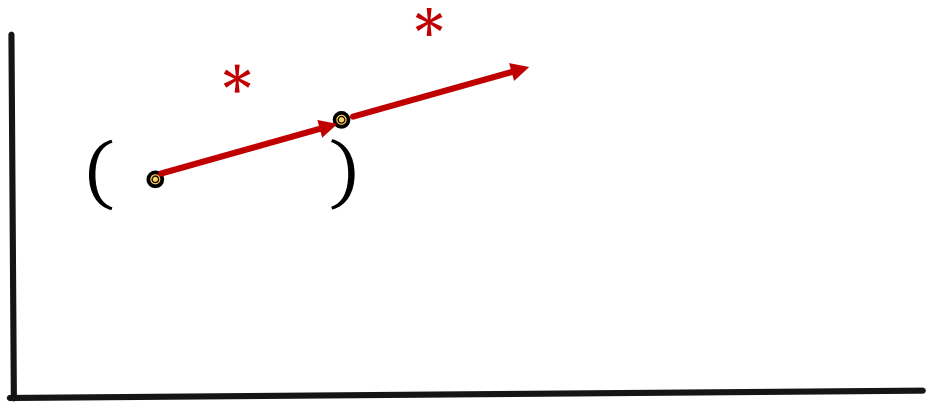
  - **Example**: # is "StudentsOf"

Antisymmetric Relations in TransE

- **Antisymmetric Relations:**

$$\$(\hbar, \%) \Rightarrow \neg\$(\%, \hbar) \square\forall \hbar, \%$$

  - **Example**: Hypernym (a word with a broader meaning: poodle vs. dog)
- **TransE** can model antisymmetric relations ü
  - (+ * =), but) + * ≠ (

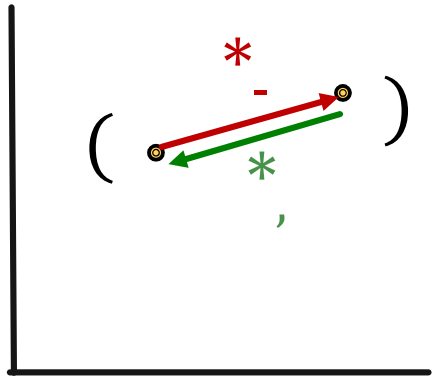


Inverse Relations in TransE

- **Inverse Relations:**

$$\$_{\#}(h, \%) \Rightarrow \$_{\$}(\%, h)$$

  - **Example** : (Advisor, Advisee)
- **TransE** can model inverse relations ü
  - (+ *, =), we can set *) = −*(



Composition in TransE

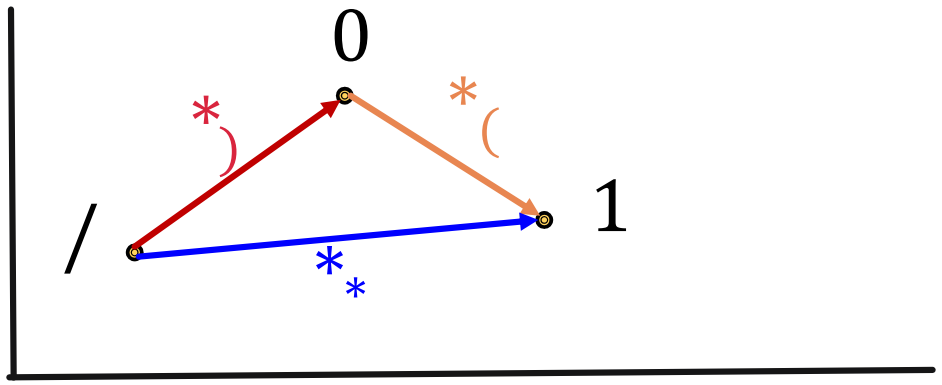
- **Composition (Transitive) Relations:**

$$\$_{\$}(7,8) \wedge \$_{\#}(8,:) \Rightarrow \$_{\%}(7,:) \quad \forall 7,8,:$$

  - **Example**: My mother's husband is my father.

- **TransE** can model composition relations

$$(_{\%} = (_{\$} + (_{\#}$$



Limitations of TransE: Symmetric Relations

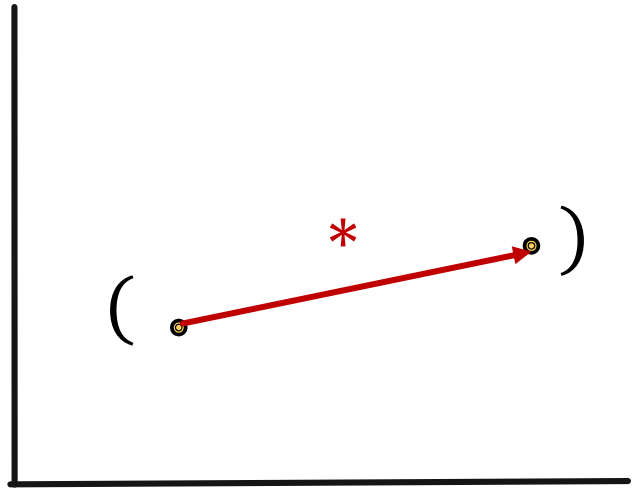
- **Symmetric Relations:**

$$\$(h, \%) \Rightarrow \$(\%, h) \square \forall h, \%$$

  - **Example**: Family, Roommate

- **TransE cannot** model symmetric relations û

only if (= 0, ' =)

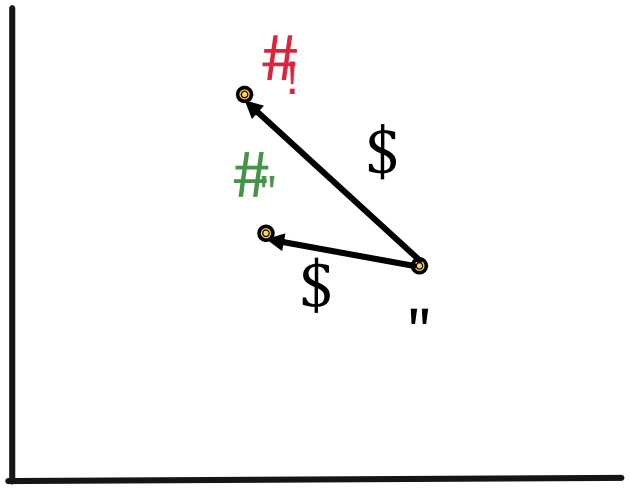


For all h, t that satisfy $r(h, t)$, $r(t, h)$ is also True, which means $\|\mathbf{h} + \mathbf{r} - \mathbf{t}\| = 0$ and $\|\mathbf{t} + \mathbf{r} - \mathbf{h}\| = 0$. Then $\mathbf{r} = 0$ and $\mathbf{h} = \mathbf{t}$, however h and t are two different entities and should be mapped to different locations.

Limitations of TransE: 1-to-N Relations

- **1-to-N Relations:**
  - **Example**: (h, r, t_1) and (h, r, t_2) both exist in the knowledge graph, e.g., r is "StudentsOf"
- **TransE cannot** model 1-to-N relations û
  - $\mathbf{t}_1$ and $\mathbf{t}_2$ will map to the same vector, although they are different entities

- $)_{\$} = ' + (=)_{\#}$
- $)_{\$} \neq)_{\#}$ contradictory!



KG Completion Methods

Model	Score	Embedding	Sym.	Antisym.	Inv.	Compos.	1-to-N
TransE	$-\|\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|\|$	$\mathbf{h},\mathbf{t},\mathbf{r}\in\mathbb{R}^{!}$	û	ü	ü	ü	û
TransR	$-\|\|\boldsymbol{M}_{"}\mathbf{h}+\mathbf{r}-\boldsymbol{M}_{"}\mathbf{t}\|\|$	$\mathbf{h},\mathbf{t}\in\mathbb{R}^{!}$, $\mathbf{r}\in\mathbb{R}^{\#}$, $\boldsymbol{M}_{"}\in\mathbb{R}^{\#\times!}$	ü	ü	ü	ü	ü
DistMult	$<\mathbf{h},\mathbf{r},\mathbf{t}>$	$\mathbf{h},\mathbf{t},\mathbf{r}\in\mathbb{R}^{!}$	ü	û	û	û	ü
ComplEx	$\mathrm{Re}(<\mathbf{h},\mathbf{r},\bar{\mathbf{t}}>)$	$\mathbf{h},\mathbf{t},\mathbf{r}\in\mathbb{C}^{!}$	ü	ü	ü	û	ü
RotateE	$-\|\|\mathbf{h}\circ\mathbf{r}-t\|\|$	$\mathbf{h},\mathbf{t},\mathbf{r}\in\mathbb{C}^{!}$	ü	ü	ü	ü	ü

Outline

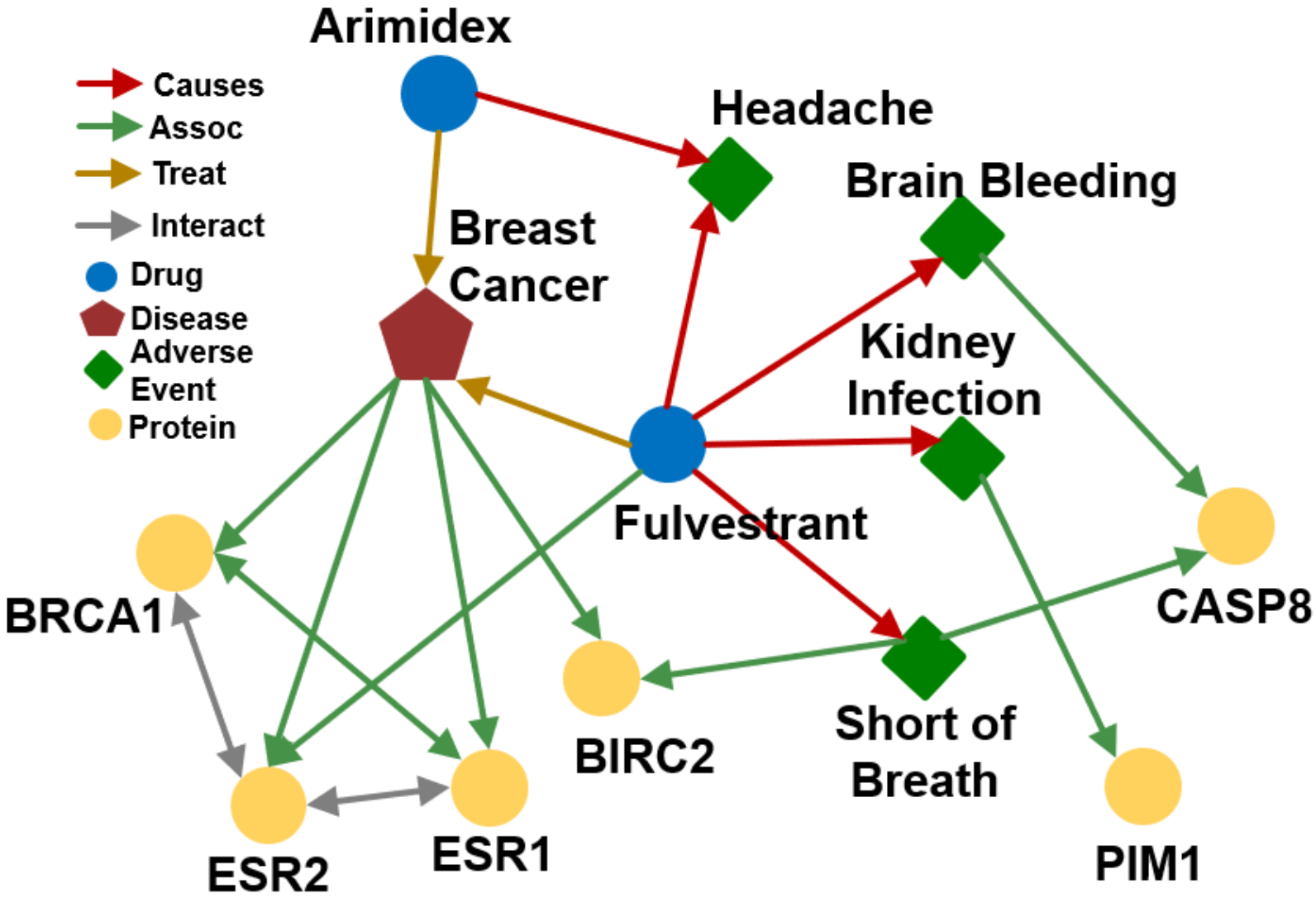
- Overview
- Knowledge Graph Completion (Link Prediction)
- **Reasoning on Knowledge Graphs**

Reasoning over KGs

- **Goal**:
  - How to perform multi-hop reasoning over KGs?

- **Reasoning over Knowledge Graphs**
  - Answering multi-hop queries
    - Path Queries
    - Conjunctive Queries
  - Query2Box

Example KG: Biomedicine



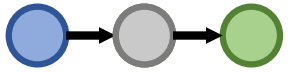
Predictive Queries on KG

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

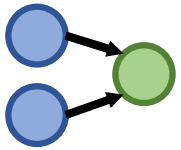
Query Types	Examples: Natural Language Question, Query
One-hop Queries	What adverse event is caused by Fulvestrant? (e:Fulvestrant, (r:Causes))
Path Queries	What protein is associated with the adverse event caused by Fulvestrant? (e:Fulvestrant, (r:Causes, r:Assoc))
Conjunctive Queries	What is the drug that treats breast cancer and caused headache? ((e:BreastCancer, (r:TreatedBy)), (e:Migraine, (r:CausedBy))

In this lecture, we only focus on answering queries on a KG!
The notation will be detailed next.

One-hop Queries



Path Queries

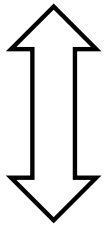


Conjunctive Queries

Predictive One-hop Queries

- **We can formulate knowledge graph completion problems as answering one-hop queries.**

- **KG completion:** Is link $(h, \$, \%)$ in the KG?



- **One-hop query:** Is $\%$ an answer to query $(h, \$)$?
  - **For example:** What side effects are caused by drug Fulvestrant?

Path Queries

- Generalize one-hop queries to path queries by **adding more relations on the path**.
- An n-hop path query q can be represented by

$$q = (v_1, (r_1, \ldots, r_n))$$

  - v_1 is an "anchor" entity,
  - Let answers to q in graph G be denoted by $[\![q]\!]_G$.

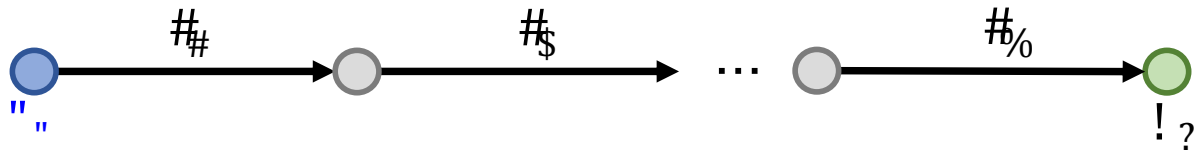
Path Queries

- Generalize one-hop queries to path queries by **adding more relations on the path**.
- An n-hop path query q can be represented by

$$q = (v_1, (r_1, \ldots, r_n))$$

  - v_1 is an "anchor" entity,
  - Let answers to q in graph G be denoted by $[\![q]\!]_G$.

Query Plan of q:

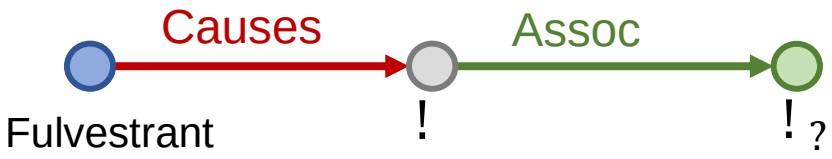


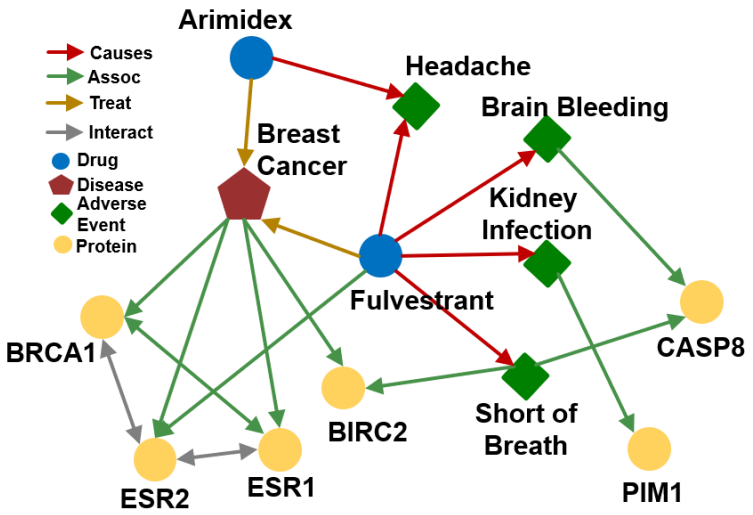
Query plan of path queries is a chain.

Path Queries

Question: "*What proteins are **associated** with adverse events **caused** by **Fulvestrant**?*"

- $*_!$ is **e:Fulvestrant**
- $(\$_', \$_\$)$ is (**r:Causes**, **r:Assoc**)
- **Query:** (**e:Fulvestrant**, (**r:Causes**, **r:Assoc**))



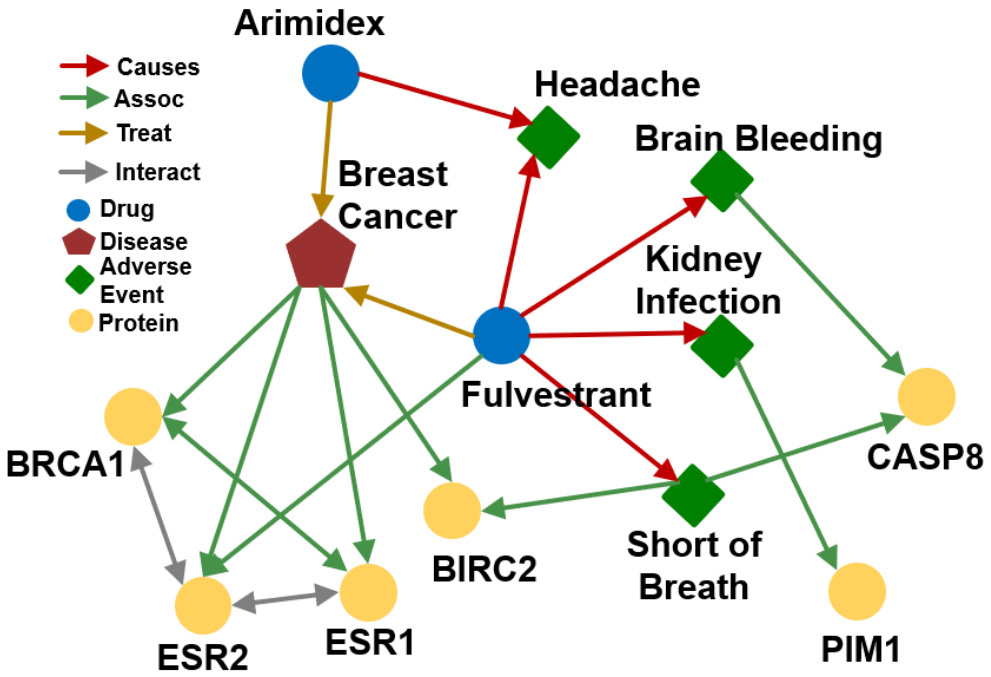


Path Queries

Question: "*What proteins are **associated** with adverse events **caused** by **Fulvestrant**?*"

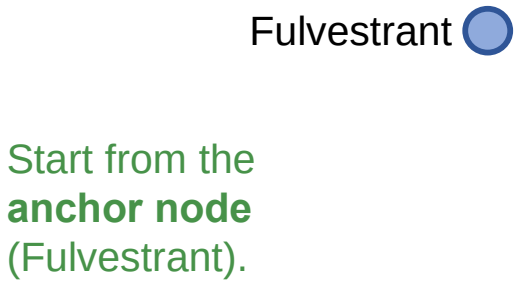
- **Query:** (e:Fulvestrant, (r:Causes, r:Assoc))

Given a KG, how to answer a path query?



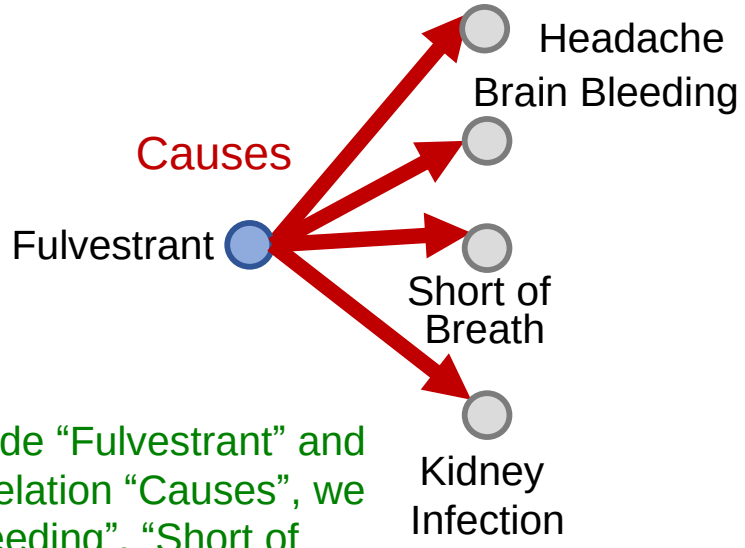
Traversing Knowledge Graphs

- **We answer path queries by traversing the KG:**
  *"What proteins are **associated** with adverse events **caused** by **Fulvestrant**?"*
- **Query:** (**e:Fulvestrant**, (**r:Causes**, **r:Assoc**))



Traversing Knowledge Graphs

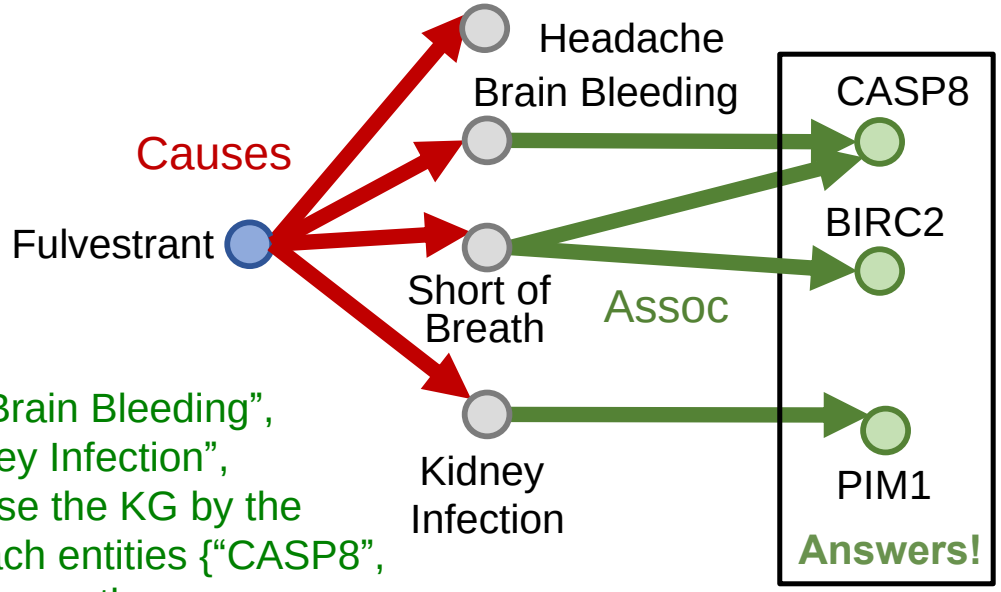
- **We answer path queries by traversing the KG:** *"What proteins are **associated** with adverse events **caused** by **Fulvestrant**?"*
- **Query: (e:Fulvestrant, (r:Causes, r:Assoc))**



Start from the anchor node "Fulvestrant" and traverse the KG by the relation "Causes", we reach entities {"Brain Bleeding", "Short of Breath", "Kidney Infection", "Headache"}.

Traversing Knowledge Graphs

- **We answer path queries by traversing the KG:**
  *"What proteins are **associated** with adverse events **caused** by **Fulvestrant**?"*
- **Query: (e:Fulvestrant, (r:Causes, r:Assoc))**



Start from the nodes {"Brain Bleeding", "Short of Breath", "Kidney Infection", "Headache"} and traverse the KG by the relation "Assoc", we reach entities {"CASP8", "BIRC2", "PIM1"}. These are the answers.

However, KGs are incomplete

- **Answering queries seems easy: Just traverse the graph.**
- **But KGs are incomplete and unknown:**
  - Many relations between entities are missing or are incomplete
    - For example, we lack all the biomedical knowledge
    - Enumerating all the facts takes non-trivial time and cost, we cannot hope that KGs will ever be fully complete
- **Due to KG incompleteness, one is not able to identify all the answer entities**

Can KG Completion Help?

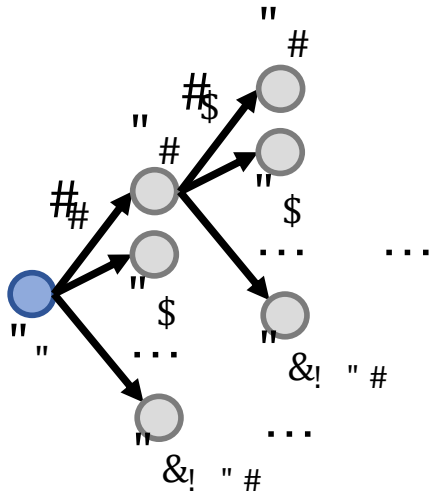
Can we first do KG completion and then traverse the completed (probabilistic) KG?

Can KG Completion Help?

Can we first do KG completion and then traverse the completed (probabilistic) KG?

- **No!** The "completed" KG is a **dense graph**!
  - Most (h, r, t) triples (edge on KG) will have some non-zero probability.

- Time complexity of traversing a dense KG is exponential as a function of the path length , : - (. ' %! &)

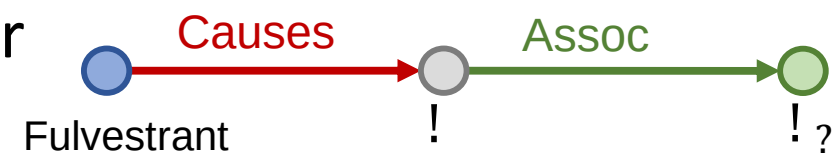


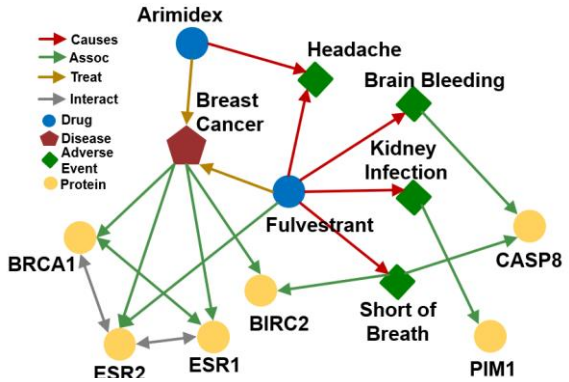
Task: Predictive Queries

- We need a way to answer path-based queries over an incomplete knowledge graph.
- We want our approach to implicitly impute and account for the incomplete KG.

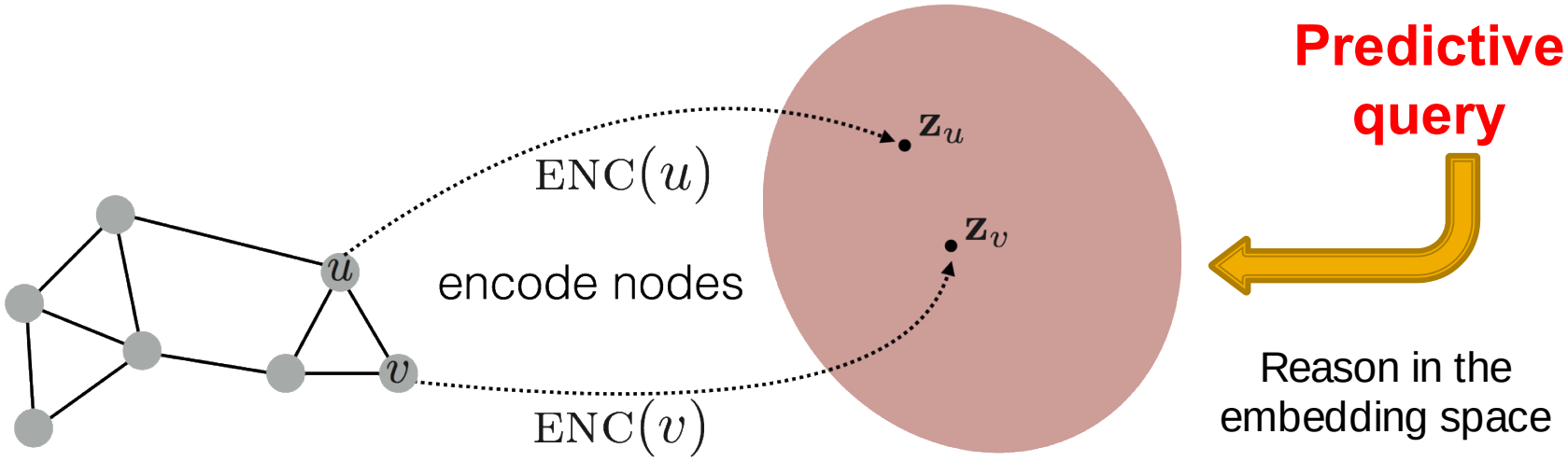
Task: Predictive Queries

- We need a way to answer path-based queries over an incomplete knowledge graph.
- We want our approach to implicitly impute and account for the incomplete KG.
- **Task: Predictive queries**
  - Want to be able to answer arbitrary queries while implicitly imputing for the missing information
  - **Generalization of the link prediction task**





A General Idea



Map queries into embedding space. **Learn to reason in that space**

- Embed query into a single **point** in the Euclidean space: answer nodes are close to the query.
- **Query2Box:** Embed query into a hyper-rectangle (**box**) in the Euclidean space: answer nodes are enclosed in the box.

[Embedding Logical Queries on Knowledge Graphs. Hamilton, et al., NeurIPS 2018]
[Query2box: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings. Ren, et al., ICLR 2020]

Traversing KG in Vector Space

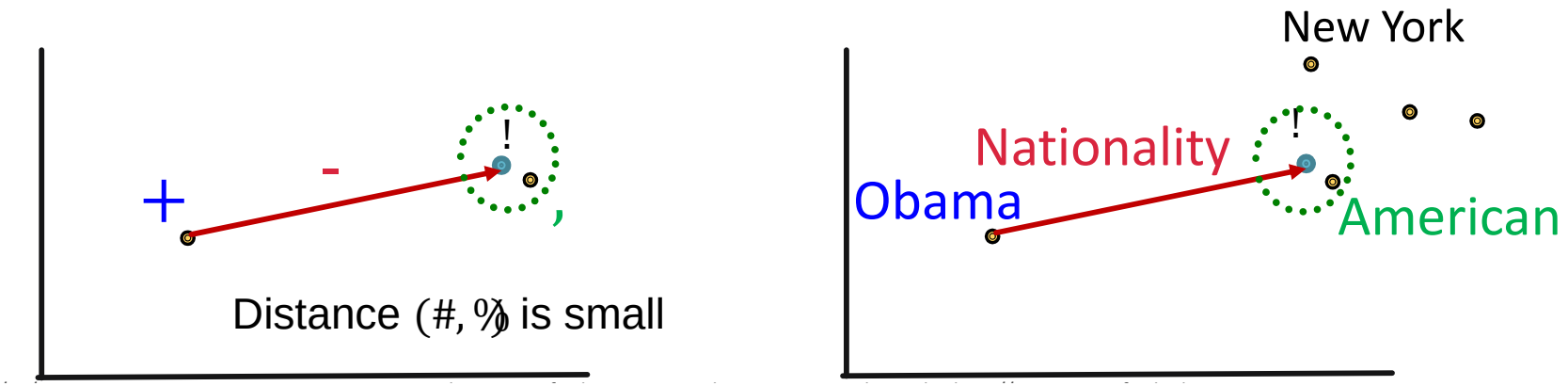
- **Key idea: Embed queries!**
  - Generalize **TransE** to multi-hop reasoning.

Guu, et al., Traversing knowledge graphs in vector space, EMNLP 2015.

Traversing KG in Vector Space

- **Key idea: Embed queries!**
  - Generalize **TransE** to multi-hop reasoning.
  - **Recap: TransE:** Translate h to t using r with score function $f(h, t) = -||h + r - t||$.
  - Another way to interpret this is that:
    - **Query embedding**: $\mathbf{q} = \mathbf{h} + \mathbf{r}$
    - Goal: **query embedding** $\mathbf{q}$ is **close** to the **answer embedding** $\mathbf{t}$

$$f(t) = -\|\mathbf{q} - \mathbf{t}\|$$

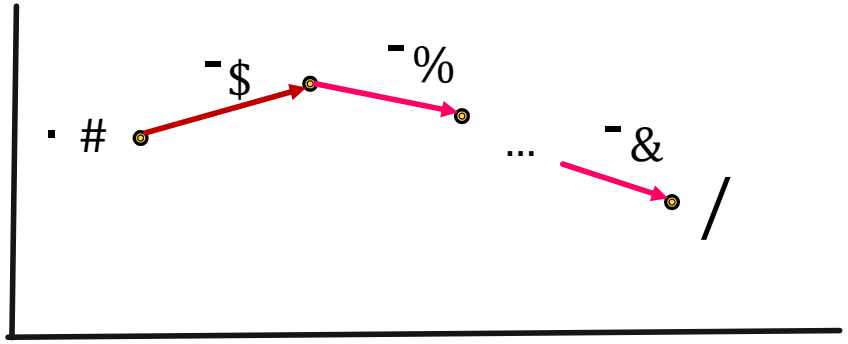


Guu, et al., Traversing knowledge graphs in vector space, EMNLP 2015.

Traversing KG in Vector Space

- **Key idea: Embed queries!**
  - Generalize **TransE** to multi-hop reasoning.

Given a path query $q = (e_0, (r_1, \ldots, r_n))$,

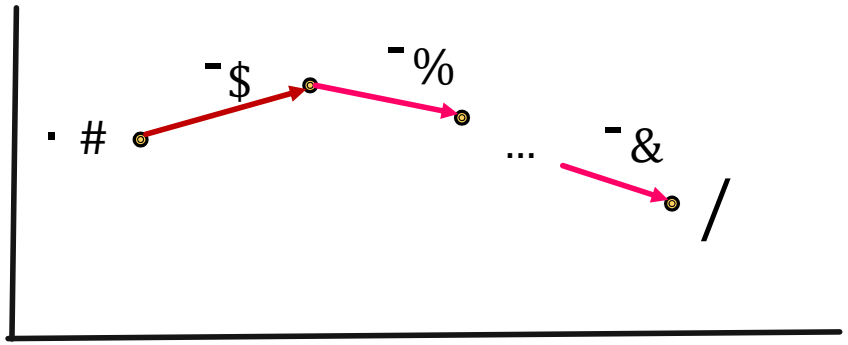


$$q = e_1 + r_1 + r_2 + \cdots + r_n$$

Traversing KG in Vector Space

- **Key idea: Embed queries!**
  - Generalize **TransE** to multi-hop reasoning.

Given a path query $q = (e_0, (r_1, \ldots, r_k))$,



$$q = e_0 + r_1 + \cdots + r_k$$

- The embedding process **only involves vector addition**, **independent of # entities** in the KG!

Traversing KG in Vector Space

Embed path queries in vector space.

- **Question:** *"What proteins are **associated** with adverse events **caused** by **Fulvestrant**?"*
- **Query:** (**e:Fulvestrant**, (**r:Causes** , **r:Assoc**))

Follow the query plan:

Query Plan **Embedding Process**

Guu, et al., Traversing knowledge graphs in vector space, EMNLP 2015.

Traversing KG in Vector Space

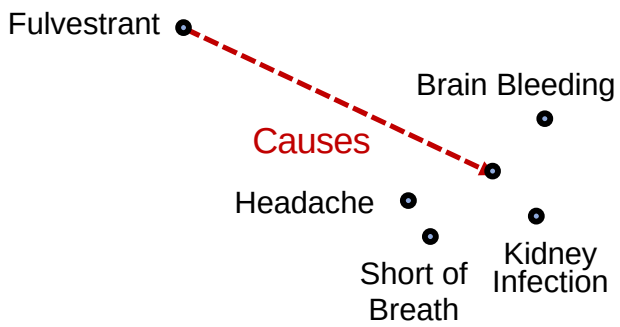
Embed path queries in vector space.

- **Question:** "*What proteins are **associated** with adverse events **caused** by **Fulvestrant**?*"
- **Query:** (**e:Fulvestrant**, (**r:Causes** , **r:Assoc**))

Follow the query plan:

Query Plan

Embedding Process



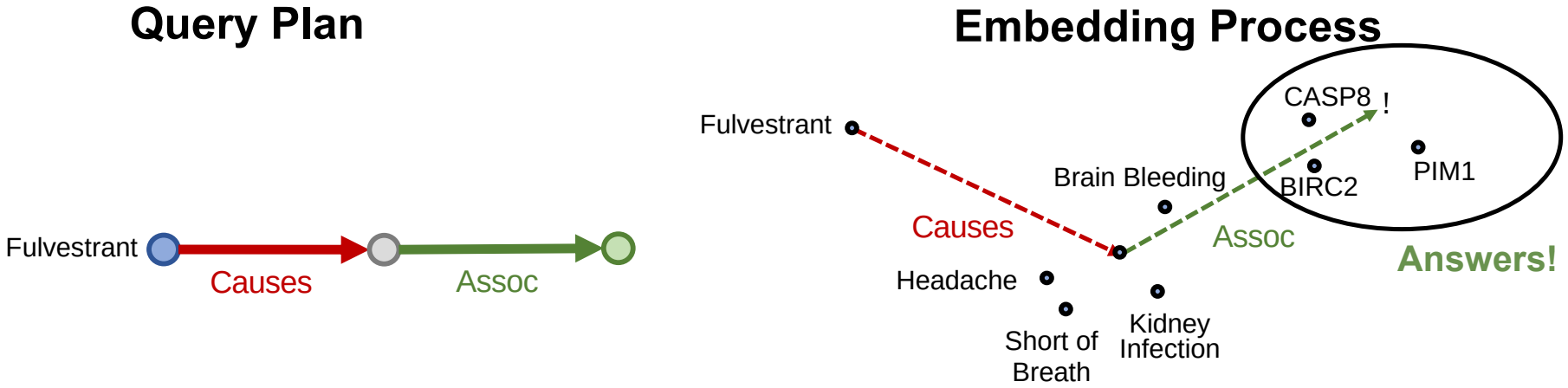
Guu, et al., Traversing knowledge graphs in vector space, EMNLP 2015.

Traversing KG in Vector Space

Embed path queries in vector space.

- **Question:** "*What proteins are **associated** with adverse events **caused** by **Fulvestrant**?*"
- **Query:** (**e:Fulvestrant**, (**r:Causes** , **r:Assoc**))

Follow the query plan:



Guu, et al., Traversing knowledge graphs in vector space, EMNLP 2015.

Traversing KG in Vector Space

Insights:

- We can train **TransE** to optimize knowledge graph completion objective

- Since **TransE** can naturally handle **compositional relations**, it can handle path queries by translating in the latent space **for multiple hops using addition of relation embeddings**.

Questions?