Knowledge Graph Embedding

Presenter: Zhiying Jiang

Agenda

- Overview
  - What & Why - Knowledge Graph
  - What & Why - Knowledge Graph Embedding
- Methods with Strengths & Weakness
  - KG Embedding with facts alone
  - KG Embedding incorporating additional Information

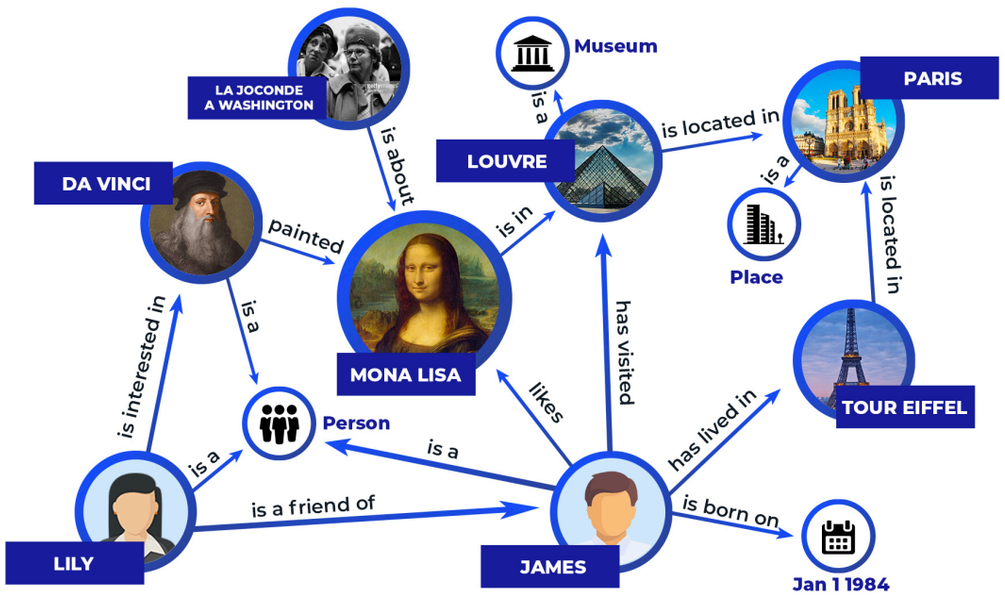
Before We Start

- This presentation is meant to be an overview of a certain topic/sub-field
- It’s not about one specific papers
- I will mainly focus on methods
- This presentation does not include a complete list of methods but some representative ones

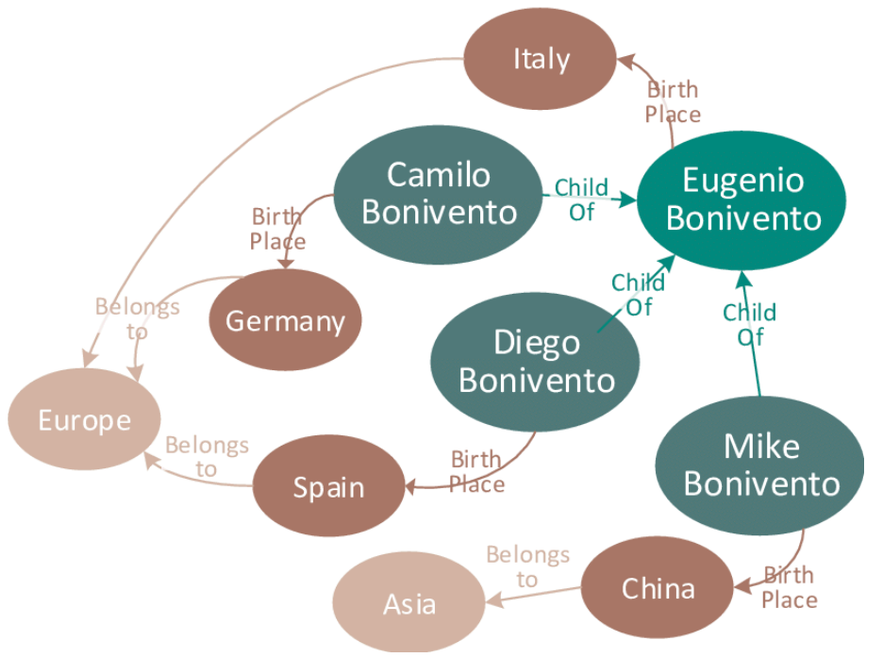
Overview

- **What & Why - Knowledge Graph**
  - **What is Knowledge Graph**

    Knowledge Graph (KG) is a multi-relational graph composed of entities (represented as nodes) and relations (represented as edges)



https://yashuseth.blog/2019/10/08/introduction-question-answering-knowledge-graphs-kgqa/



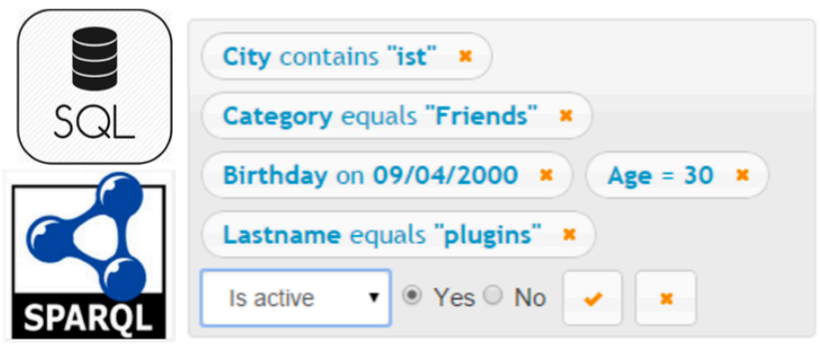
https://www.researchgate.net/figure/Motivating-Example-A-portion-of-a-Knowledge-Graph-KG-describing-relationships-among_fig1_318138527

Overview

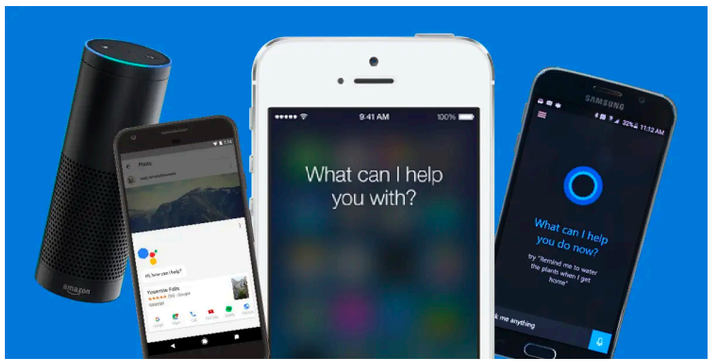
- **What & Why - Knowledge Graph**
  - **Why do we need Knowledge Graph**

    It's turning unstructured text data into structured graph data, so that textual data can become knowledge & insights :

Structured Search & Exploration



Dialog Systems



Question Answering

Passage Sentence

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity.

Question

What causes precipitation to fall?

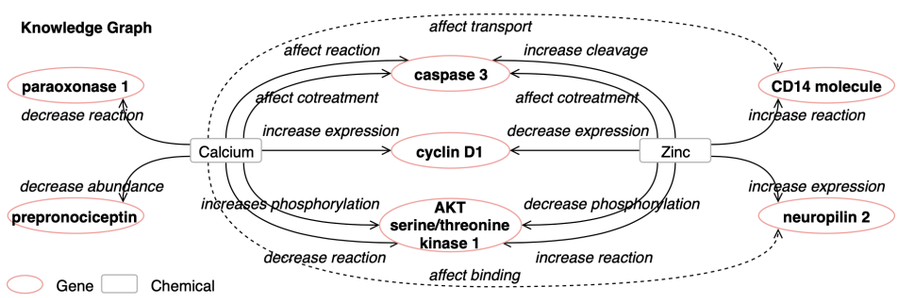
Answer Candidate

gravity

- Between question and answer

cause---gravity
precipitation---gravity
fall---gravity
what---gravity

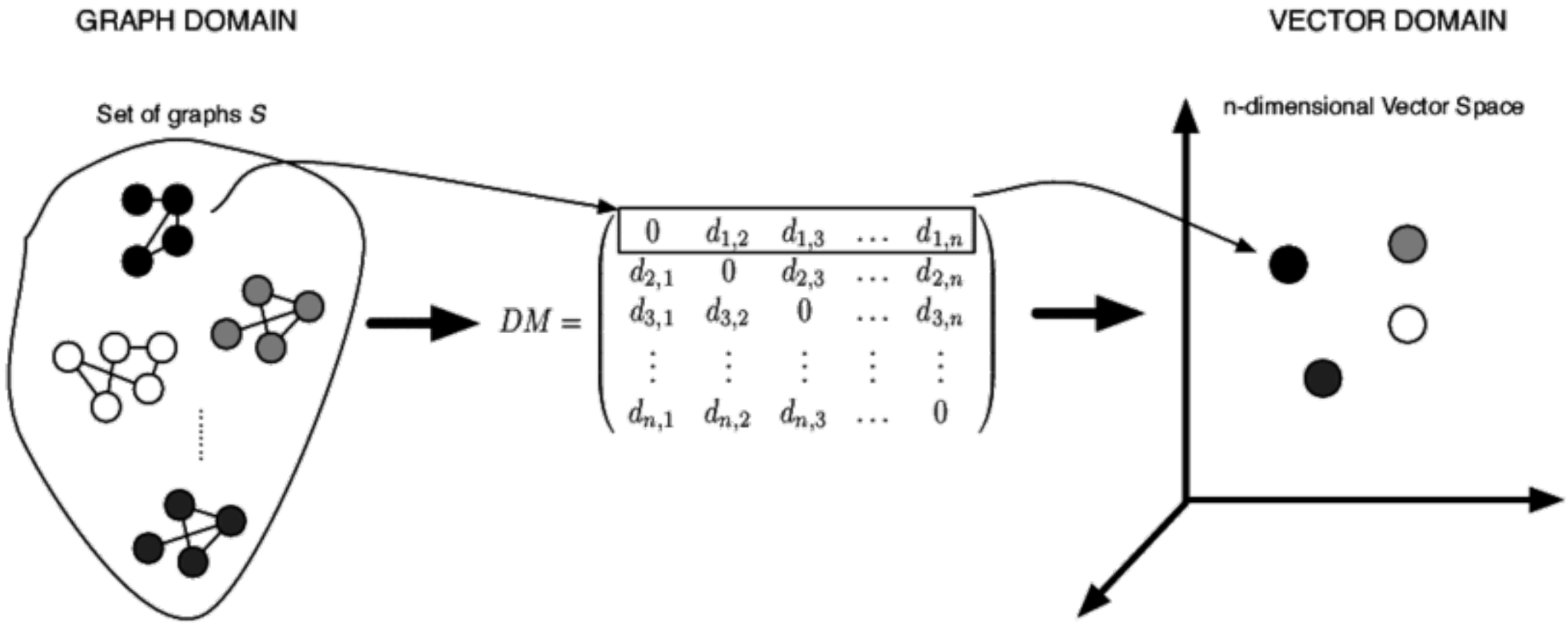
Scientific Inference



Overview

- **What & Why - Knowledge Graph Embedding**
  - **What is Knowledge Graph Embedding**

    Knowledge Graph (KG) embedding is to embed components (entities, relations) of KG into continuous vector space



https://www.researchgate.net/figure/Illustration-of-the-first-step-graph-embedding_fig2_221275649

Overview

- **What & Why - Knowledge Graph Embedding**
  - **Why do we need Knowledge Graph Embedding**
    - To simplify the manipulation while preserving the inherent structure of the KG
    - To benefit downstream tasks such as KG completion, relation extraction, entity classification, and entity resolution

e.g., KG Completion

KG is always represented in millions of triples: (entity1, relation, entity2), but it's not complete - there are a lot of missing links. So the goal of this task is to predict the missing part of the triple, which can be :

- ✦ Given entity1 and entity2, predict relation
- ✦ Given entity1 and relation, predict the missing entity2
- ✦ Given a triple, predict whether it's true or false

Methods

- **KG Embedding with facts alone**

  Before digging into different methods, let's define the task formally:

  Given a KG consisting of n entities and m relations, and facts observed in the KG are stored as a collection of triples - $\mathbb{D}^+ = \{(h, r, t)\}$

  where $h \in \mathbb{E}, t \in \mathbb{E}, r \in \mathbb{R}$. $\mathbb{E}$ represent the set of entities, $\mathbb{R}$ represents the set of relations
  (h can be called as head entity, t can be called as tail entity; or they can all be called as node in different contexts)

  e.g., (`DavidFincher, DirectorOf, FightClub`)

Methods

- **KG Embedding with facts alone**

  - Common Procedures on a High Level:

  - Represent entities and relations
    - Options are vectors, matrices, tensors, or modelling them through multivariate Gaussian distributions
  - Define a scoring function
    - To measure the plausibility of a fact. That’s the part where methods vary from one to another
  - Learn entity and relation representations
    - Through maximizing the total plausibility of observed facts

Methods

- **KG Embedding with facts alone**
  - **We can roughly categorize those methods into 3 directions**
  - **For each direction, I will choose one representative paper and introduce them in different detailed levels**

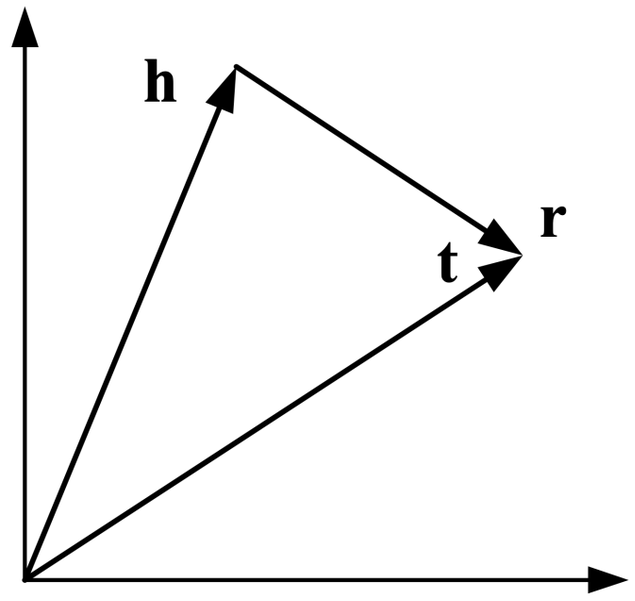
    - Translation-Based Models
      - TransE [Bordes et al. NeurIPS 2013.]
    - Semantic Matching Models
      - RESCAL [Nickel et al. ICML 2011.]
    - Graph Based Models
      - Graph Attention Networks [Veličković et al. ICLR 2018.]

TransE

- The very first paper of Translation-Based Model
- It opens a whole new direction and gives people a whole new perspective of knowledge graph embedding

- What is the key idea of translation-based model?

It assumes - in the vector space, when adding the relation to the head entity, we should get close to the target tail entity.



Entity and Relation Space

Given a fact (h, r, t), we will have

$$h + r \approx t$$

Then the score function is obvious

$$f_r(h, t) = - \parallel h + r - t \parallel$$

TransE

- Given the score function, how can we train the model?

- We are given a KG, and we know all of triples include in the KG are facts, which means they are true (positive)
- Then we need some negative samples to compare the facts with so that we can train the model to learn the embeddings
- There are 2 different assumptions when training, which affect the definition of “negative samples”
  - Open World Assumption
    ➡ Unobserved triples are either wrong or missing
  - Close Word Assumption
    ➡ Unobserved triples are all wrong
- We can then generate negative samples, e.g., we can replace head/tail entity with a random head/tail entities, or replace relation with other relations (of course there are other more complex negative sampling methods)

TransE

- Given the score function, how can we train the model?

- Suppose now we have positive sample (h, r, t) and negative sample (h', r', t'), pairwise ranking loss is often used as the loss function under Open World Assumption:

$$L = max(0, \gamma - f_r(h, t) + f_{r'}(h', t'))$$

L is loss for single pair of positive and negative sample

γ is the margin between positive and negative score functions

- Then for the whole training set, we will have:

$$\mathscr{L} = min \sum_{(h,r,t)\in\mathbb{D}^+} \sum_{(h',r',t')\in\mathbb{D}^-} max(0, \gamma - f_r(h, t) + f_{r'}(h', t'))$$

$\mathbb{D}^+$ is the set of positive triples while $\mathbb{D}^-$ is the set of negative triples

- The goal is to make the positive triple achieve higher “score” than negative triples

TransE

- In a nutshell, the training procedure is as follows:

Algorithm 1 Training under Open World Assumption

Input: Observed facts $\mathbb{D}^+ = \{(h, r, t)\}$

Initialize entity and relation embeddings
loop
  $\mathbb{P} \leftarrow$ a small set of positive facts sampled from $\mathbb{D}^+$
  $\mathbb{B}^+ \leftarrow \emptyset, \ \ \mathbb{B}^- \leftarrow \emptyset$
  foreach $\tau^+ = (h, r, t) \in \mathbb{P}$ **do**
    Generate a negative fact $\tau^- = (h', r', t')$
    $\mathbb{B}^+ \leftarrow \mathbb{B}^+ \cup \{\tau^+\}, \ \ \mathbb{B}^- \leftarrow \mathbb{B}^- \cup \{\tau^-\}$
  end for
  Update entity and relation embeddings w.r.t. the gradients of $\sum_{\tau \in \mathbb{B}^+ \cup \mathbb{B}^-} \log\left(1 + \exp(-y_{hrt} \cdot f_r(h, t))\right)$ or $\sum_{\tau^+ \in \mathbb{B}^+, \tau^- \in \mathbb{B}^-} \max\left(0, \gamma - f_r(h, t) + f_{r'}(h', t')\right)$
  Handle additional constraints or regularization terms
end loop

Output: Entity and relation embeddings

TransE

- What's the strength and weakness of this method?

- It is simple and efficient
- The performance is really good comparing to previous methods
- But it can not handle one-to-multiple, multiple-to-one, and multiple-to-multiple relations well
- In general, this method is very innovative in terms of how they model the embedding problems and auspicate a whole new direction (see the image below)

Method	Ent. embedding	Rel. embedding	Scoring function $f_r(h,t)$	Constraints/Regularization
TransE [14]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$-\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{1/2}$	$\|\mathbf{h}\|_2=1, \|\mathbf{t}\|_2=1$
TransH [15]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r},\mathbf{w}_r\in\mathbb{R}^d$	$-\|(\mathbf{h}-\mathbf{w}_r^\top\mathbf{h}\mathbf{w}_r)+\mathbf{r}-(\mathbf{t}-\mathbf{w}_r^\top\mathbf{t}\mathbf{w}_r)\|_2^2$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1$ $\|\mathbf{w}_r^\top\mathbf{r}\|/\|\mathbf{r}\|_2\le\epsilon, \|\mathbf{w}_r\|_2=1$
TransR [16]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^k, \mathbf{M}_r\in\mathbb{R}^{k\times d}$	$-\|\mathbf{M}_r\mathbf{h}+\mathbf{r}-\mathbf{M}_r\mathbf{t}\|_2^2$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$ $\|\mathbf{M}_r\mathbf{h}\|_2\le 1, \|\mathbf{M}_r\mathbf{t}\|_2\le 1$
TransD [50]	$\mathbf{h},\mathbf{w}_h\in\mathbb{R}^d$ $\mathbf{t},\mathbf{w}_t\in\mathbb{R}^d$	$\mathbf{r},\mathbf{w}_r\in\mathbb{R}^k$	$-\|(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}+\mathbf{r}-(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\|_2^2$	$\|\mathbf{h}\|_2\le 1, \|\mathbf{t}\|_2\le 1, \|\mathbf{r}\|_2\le 1$ $\|(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}\|_2\le 1$ $\|(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\|_2\le 1$

Small portions of translation-based model

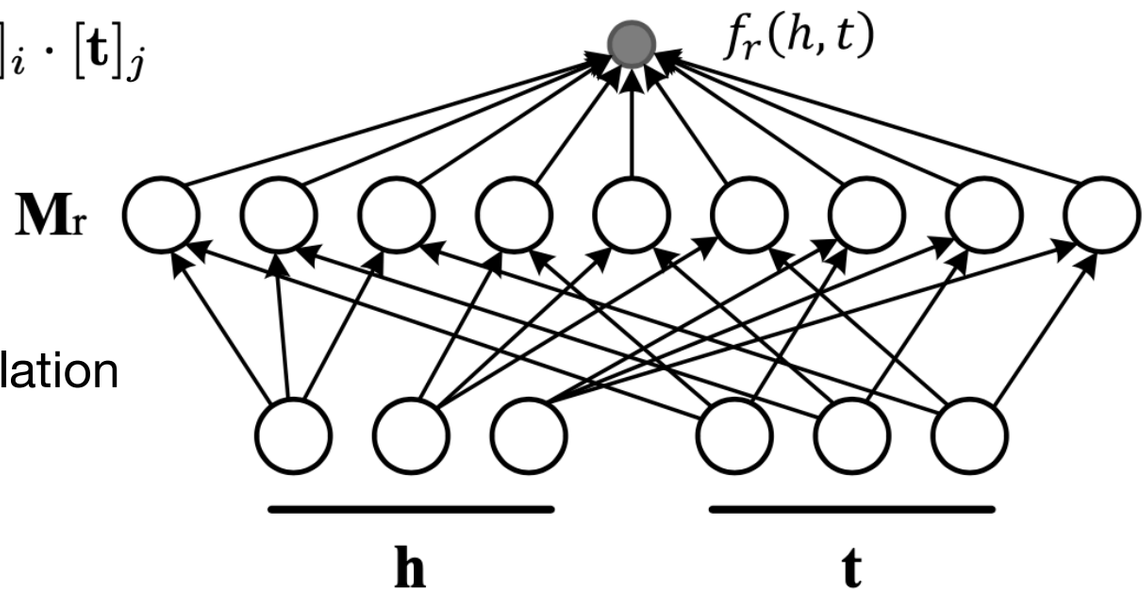
Wang, Quan, et al. "Knowledge graph embedding: A survey of approaches and applications." *IEEE Transactions on Knowledge and Data Engineering* 29.12 (2017): 2724-2743.

RESCAL

- The very first latent feature model for knowledge graph embedding
- RESCAL associates each entity with a vector to capture its latent semantics
- Each relation is represented as a matrix which models models pairwise interactions between latent factors
- The score function is defined as:

$$f_r(h,t) = \mathbf{h}^\top \mathbf{M}_r \mathbf{t} = \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} [\mathbf{M}_r]_{ij} \cdot [\mathbf{h}]_i \cdot [\mathbf{t}]_j$$

$\mathbf{M}_r \in \mathbb{R}^{d \times d}$ is a matrix associated with relation



Graph Attention Networks

- The very first paper to utilize Self-Attention in graph embedding

  - What is Graph Attention Networks?

  - The main idea is to enable neighbours of one node to attend representation with different weights
  - The key element is its Graph Attention Networks (GAT) layer
  - If we take an analogy to transformer, each node is treated as the word to embed, and selected neighbours are treated as context words
  - Formally, we define a set of node features, $h = \{h_1, h_2, \ldots, h_N\}, h_i \in \mathbb{R}^F$, where N is the number of nodes, and F is the number of features in each node
  - The output of GAT layer will be $h' = \{h'_1, h'_2, \ldots, h'_N\}, h'_i \in \mathbb{F}'$

Graph Attention Networks

- What is Graph Attention Networks in detail?

- Define a weight matrix, $\mathrm{W} \in \mathbb{R}^{F' \times F}$, applied to every node

  ➡ $\mathrm{W}h_i$

- Perform Self-Attention $a : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \rightarrow \mathbb{R}$ among nodes

  ➡ $e_{ij} = a(\mathrm{W}h_i, \mathrm{W}h_j)$, where $j \in N_i$, N_i is neighbours of i

- Normalize coefficients across all choices of h_j using softmax function

  ➡ $\alpha_{ij} = softmax(e_{ij}) = \dfrac{exp(e_{ij})}{\sum_{k \in N_i} exp(e_{ik})}$

- Represent h_i' as linear combination of its neighbours

  ➡ $h_i' = \sigma(\sum_{j \in N_i} \alpha_{ij} \mathrm{W}h_j)$

Graph Attention Networks

- What is Graph Attention Networks in more detail?

- In the paper, Self-Attention $a : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \rightarrow \mathbb{R}$ is a single-layer feedforward neural network with LeakyReLU as nonlinear function

  ➡ $\alpha_{ij} = \frac{exp(LeakyReLU(a[Wh_i \parallel Wh_j]))}{\sum_{k \in N_i} exp(LeakyReLU(a[Wh_i \parallel Wh_j]))}$, where $\parallel$ is concatenation operation

- Also multi-head attention is used in this paper, so we have

  ➡ $h_i' = \Big\|_{k=1}^{K} \sigma\Big(\sum_{j \in N_i} \alpha_{ij} \mathrm{W} h_j\Big)$, where $\parallel$ is concatenation and K represents the number of head we have

Graph Attention Networks

- What is Graph Attention Networks in more detail?

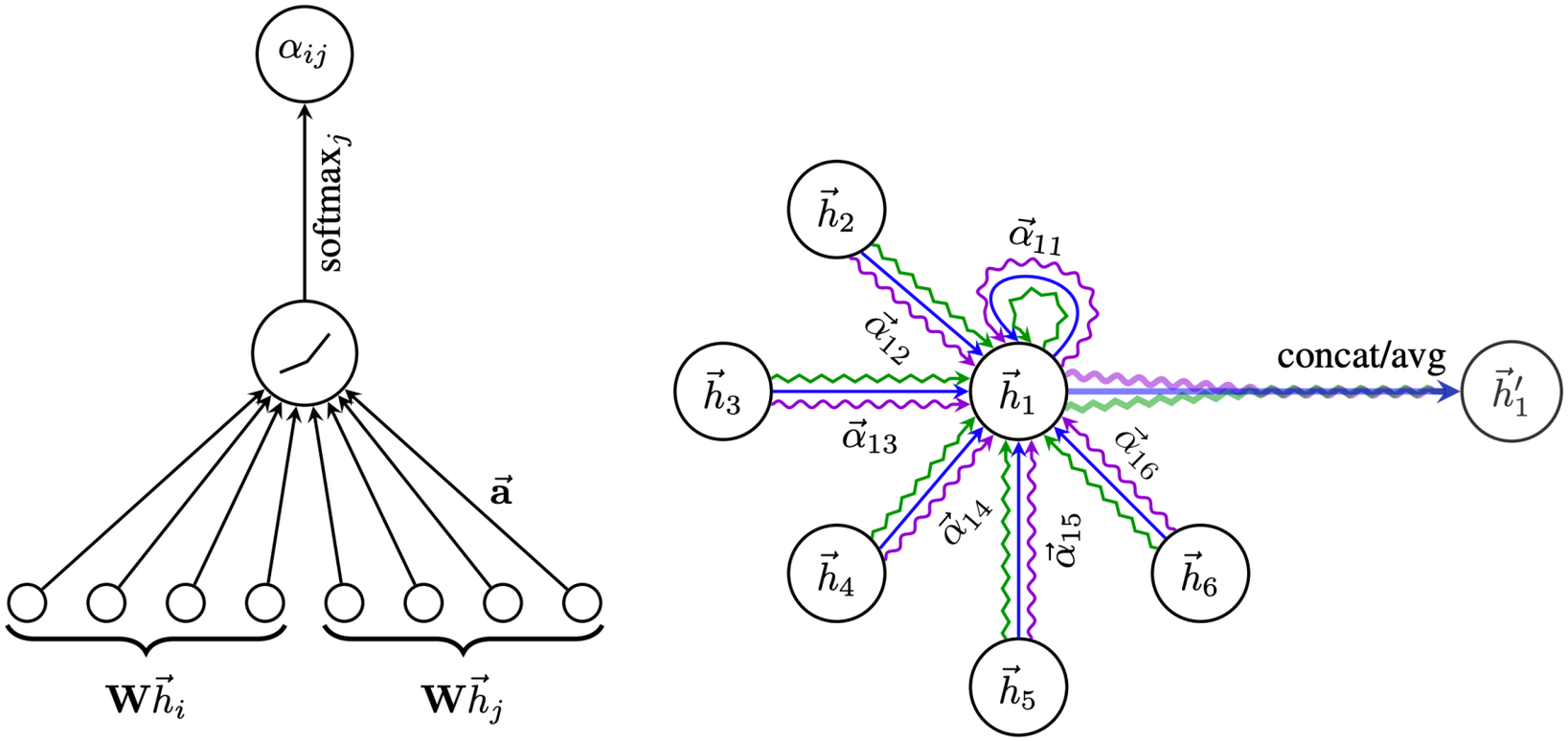


Figure 1: **Left:** The attention mechanism $a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$ employed by our model, parametrized by a weight vector $\vec{\mathbf{a}} \in \mathbb{R}^{2F'}$, applying a LeakyReLU activation. **Right:** An illustration of multi-head attention (with $K = 3$ heads) by node 1 on its neighborhood. Different arrow styles and colors denote independent attention computations. The aggregated features from each head are concatenated or averaged to obtain $\vec{h}'_1$.

Graph Attention Networks

- What's the strength and weakness of this method?

- It is computationally efficient since it doesn't require matrix operation and is parallelizable
- It allows assigning different importances to different nodes implicitly
- It doesn't require to know the entire graph structure
- It's also flexible, with only GAT layer, we can insert it into any other architectures
- It can also leverage the interpretability of knowledge graph embedding as we can have attribution map according to attention scores
- However, it only shows the strong performance in node classification problem. And the lack of considering relations may make it disadvantageous in a certain downstream tasks (e.g., link prediction)

Methods

- **KG Embedding incorporating additional information**

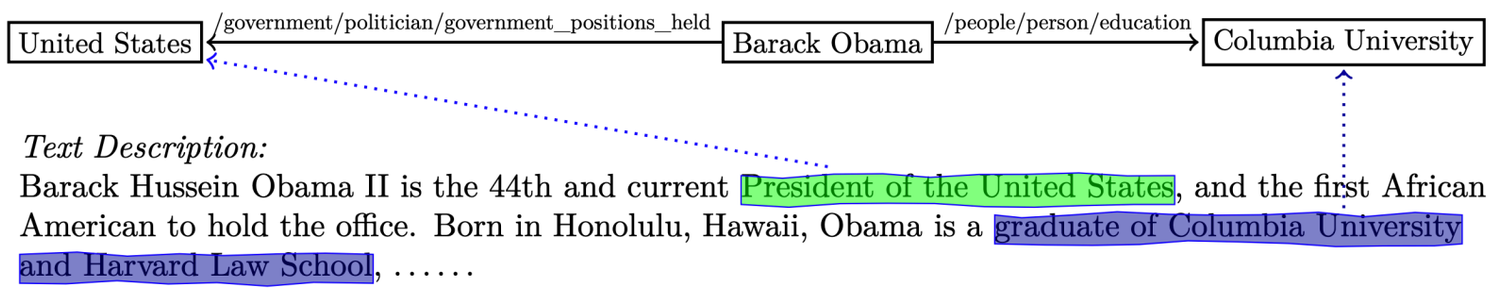
  Another stream of work utilize additional information in knowledge graph embedding, and the common additional information can be:

  - Entity Types
  - Textual Descriptions
    - Knowledge Graph Representation with Jointly Structural and Textual Encoding [Xu et al. IJCAI 2017.]
  - Relation Paths
    - Modelling Relation Paths for Representation Learning of Knowledge Bases [Lin et al. EMNLP 2015.]
  - Pretrained Language Models
    - KG-BERT: BERT for Knowledge Graph Completion [Yao et al. AAAI 2020.]

Knowledge Graph Representation with Jointly Structural and Textual Encoding

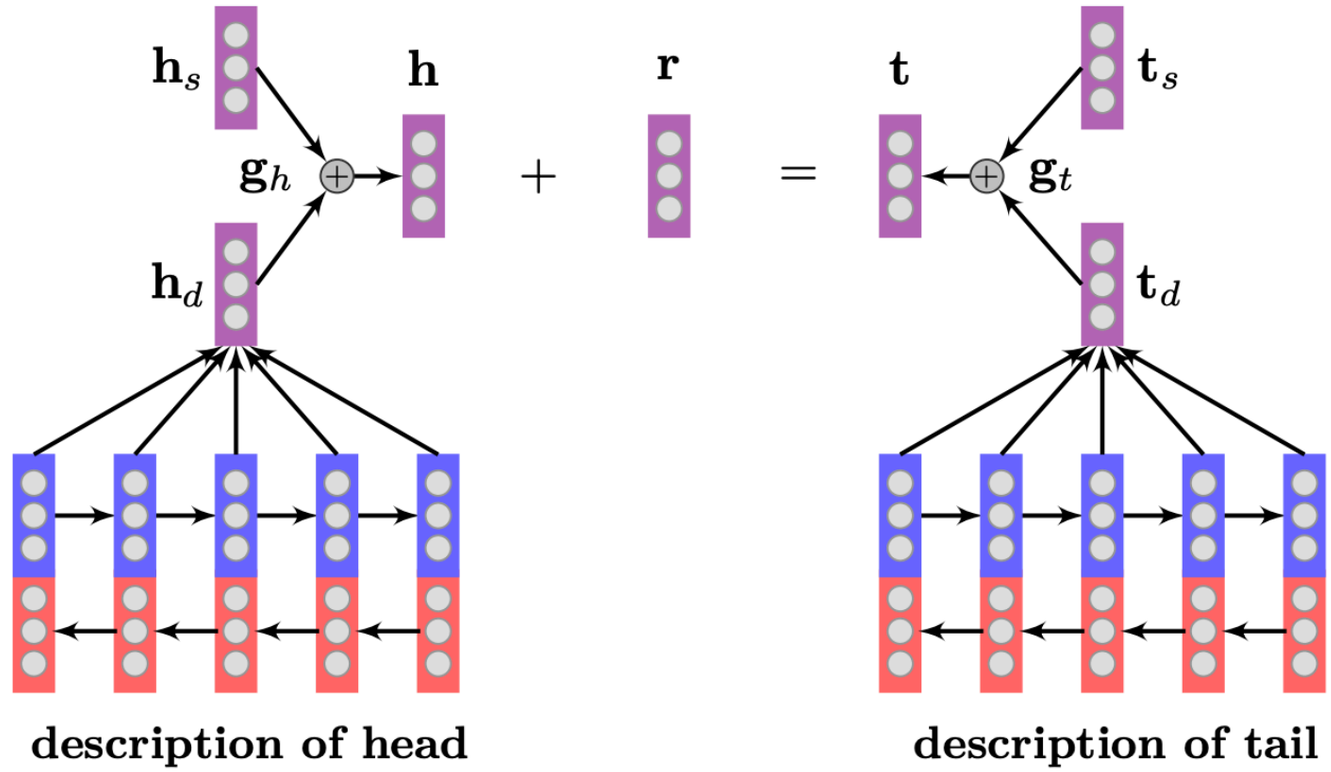
- The very first paper to incorporate both structural information and textual information in knowledge graph embedding
- It utilized entity description as the textual information

e.g.,



- They used various methods to encode textual information (Bag-of-Words, LSTM Encoder, Attentive LSTM Encoder)
- They utilized pretrained TransE embedding for structural representation
- The innovation part is that they used gated mechanism to balance between the structure information and textual information

Knowledge Graph Representation with Jointly Structural and Textual Encoding



- As we can see, the general framework is the same as TransE, and the only difference is that they incorporated gate mechanism in entity representation $\mathrm{e} = \mathrm{g}_e \odot \mathrm{e}_s + (1 - \mathrm{g}_e) \odot e_d$

Other Additional Information used in KG Embedding

- Modelling Relation Paths for Representation Learning of Knowledge Bases [Lin et al. EMNLP 2015.]
  - Same framework as TransE and encoding the path from head entity to tail entity using different strategies (addition, multiplication, RNN)
- KG-BERT: BERT for Knowledge Graph Completion [Yao et al. AAAI 2020.]
  - Fine tune BERT on different downstream tasks (triple classification, relation prediction, link prediction).
  - It's based on Close World Assumption, which assume every unobserved triple to be negative.

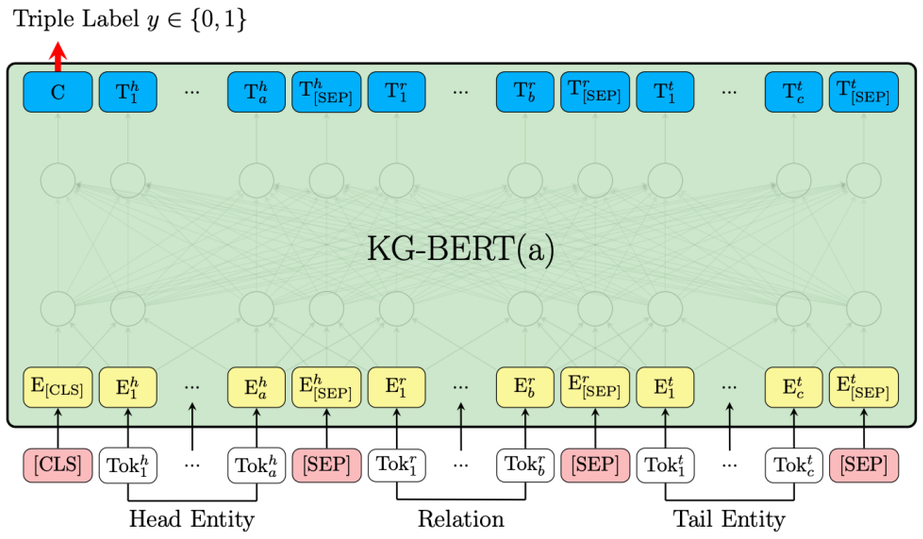


Figure 1: Illustrations of fine-tuning KG-BERT for predicting the plausibility of a triple.

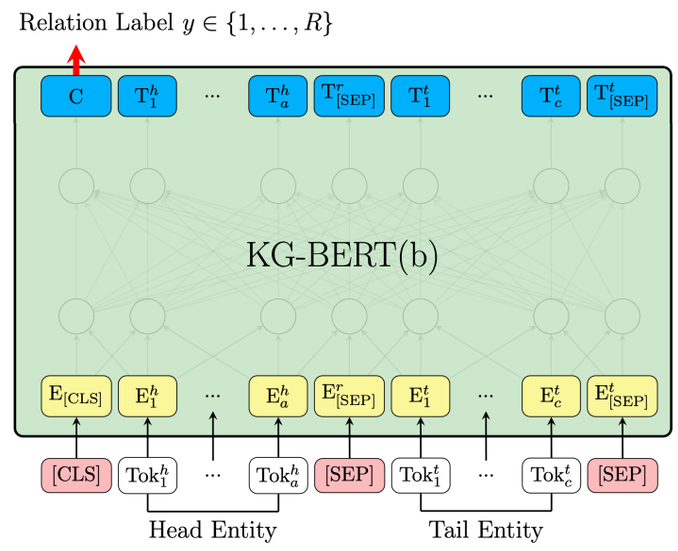


Figure 2: Illustrations of fine-tuning KG-BERT for predicting the relation between two entities.

Summary

- Knowledge Graph embedding is a trending topic that can help people discover the underlying facts, especially when more and more information in the real world is stored as structured data (wikidata, yago, freebase, dbpedia)

- In this presentation, I introduced some representative knowledge graph embedding models with different levels of detail

- Hopefully this introduction can interest people and inspire more and more people working on it :)

References

- Yao, Liang, Chengsheng Mao, and Yuan Luo. "KG-BERT: BERT for Knowledge Graph Completion." *arXiv preprint arXiv:1909.03193* (2019).
- Wang, Quan, et al. "Knowledge graph embedding: A survey of approaches and applications." *IEEE Transactions on Knowledge and Data Engineering* 29.12 (2017): 2724-2743.
- Lin, Yankai, et al. "Modeling relation paths for representation learning of knowledge bases." *arXiv preprint arXiv:1506.00379* (2015).
- Veličković, Petar, et al. "Graph attention networks." *arXiv preprint arXiv:1710.10903* (2017).
- Bordes, Antoine, et al. "Translating embeddings for modeling multi-relational data." *Advances in neural information processing systems*. 2013.
- Xu, Jiacheng, et al. "Knowledge graph representation with jointly structural and textual encoding." *arXiv preprint arXiv:1611.08661* (2016).
- Nickel, Maximilian, Volker Tresp, and Hans-Peter Kriegel. "A three-way model for collective learning on multi-relational data." *Icml*. Vol. 11. 2011.

Thank You!