KNOWGRAPH COLLOQUIUM

Knowledge graph embedding survey by Wang et al.

Jean N'dah KOUAGOU
DICE Research Group

Research Presentation
Online

15 December 2020

Embedding with facts alone

Definition

- Knowledge graph (KG) embedding is to embed all components of a KG including entities and relations into continuous vector spaces
- It simplifies the manipulation while preserving the inherent structure of the KG
- How does it work ?
  - Represent entities and relations (for instance initialize them with random vectors)
  - Define a scoring function (to mesure the plausibility of facts)
  - Learn entity and relation representations (for instance by training a neural network)
- Observed facts tend to have higher scores as compared to non observed facts

- **Two categories of embedding models**
  1. **Translational distance models**
  2. **Semantic matching models**

- **Two categories of embedding models**
  1. **Translational distance models**
  2. **Semantic matching models**

- **In the next slides, h, t and r are the vector representations of the head entity, tail entity and the relation respectively**

- **All figures are adapted from the paper**

- **Exploit distance-based scoring functions**

- **Measure the plausibility of a fact as the distance between two entities, usually after a translation carried out by the relation**

Examples (1/3): TransE

- TransE represents entities and relations as vectors
- The scoring function is:

$$f_r(h, t) = -\|h + r - t\|_{\frac{1}{2}}$$

Examples (1/3): TransE

- TransE represents entities and relations as vectors
- The scoring function is:

$$f_r(h, t) = -\|h + r - t\|_{\frac{1}{2}}$$

- Limitations of TransE: it does not support 1-to-N, N-to-N and N-to-1 relations

Examples (2/3): TransH

- TransH represents entities and relations as vectors
- It overcomes the limitations of TransE by introducing a relation-specific hyperplane with normal vector w_r
- The scoring function is:

$$f_r(h, t) = -\|h_\perp + r - t_\perp\|_2^2,$$

where $h_\perp = h - w_r^T h w_r$, $t_\perp = t - w_r^T t w_r$

Embedding with facts alone

Translational distance models – Examples

TransE vs. TransH – Visualization

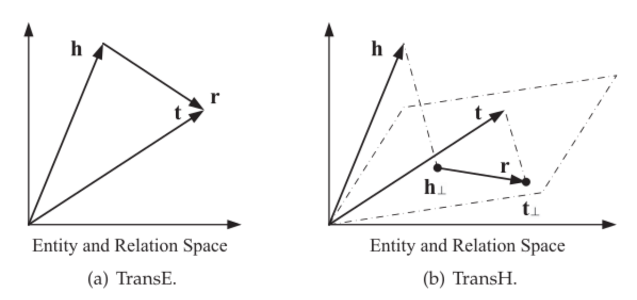

Figure: TransE vs. TransH

Examples (3/3): KG2E

- KG2E represents entities and relations as random vectors drawn from multivariate Gaussian distributions:

Examples (3/3): KG2E

- KG2E represents entities and relations as random vectors drawn from multivariate Gaussian distributions:

$$
\begin{aligned}
h &\sim \mathcal{N}(\mu_h, \Sigma_h) \\
t &\sim \mathcal{N}(\mu_t, \Sigma_t) \\
r &\sim \mathcal{N}(\mu_r, \Sigma_r)
\end{aligned}
$$

- With Gaussian embeddings, KG2E can effectively measure uncertainties of entities and relations

Examples (3/3): KG2E

- KG2E represents entities and relations as random vectors drawn from multivariate Gaussian distributions:

$$h \sim \mathcal{N}(\mu_h, \Sigma_h)$$
$$t \sim \mathcal{N}(\mu_t, \Sigma_t)$$
$$r \sim \mathcal{N}(\mu_r, \Sigma_r)$$

- With Gaussian embeddings, KG2E can effectively measure uncertainties of entities and relations
- Two ways to define the scoring function:
    1. Using the Kullback Leibler divergence
    2. Using the probability inner product

- **Exploit similarity-based scoring functions**
- **Measure plausibility of facts by matching latent semantics of entities and relations embodied in their vector space representations**

Examples (1/3) – RESCAL

- RESCAL represents each entity as a vector and each relation as a matrix

DICE

Examples (1/3) – RESCAL

- RESCAL represents each entity as a vector and each relation as a matrix
- Its scoring function is:

$$f_r(h, t) = h^T M_r t$$

- RESCAL can model pairwise interactions between all the components of h and t
- Limitations of RESCAL: scalability issues

Examples (2/3) – Distmult

- Distmult simplifies RESCAL by considering M_r to be a diagonal matrix

DICE

Examples (2/3) – Distmult

- Distmult simplifies RESCAL by considering M_r to be a diagonal matrix
- Its scoring function is:

$$f_r(h, t) = h^T diag(r) t$$

- Captures pairwise interactions between only components along the same dimension

Examples (3/3) – Neural Tensor Network (NTN)

- NTN represents both entities and relations as a vector
- Each relation is associated with a three-dimensional tensor $\underline{M}$, two weight matrices M^1 and M^2, and a bias vector b

DICE

Examples (3/3) – Neural Tensor Network (NTN)

- NTN represents both entities and relations as a vector
- Each relation is associated with a three-dimensional tensor $\underline{M}$, two weight matrices M^1 and M^2, and a bias vector b
- Its scoring function is:

$$f_r(h,t) = r^T \tanh(h^T \underline{M}_r t + M_r^1 h + M_r^2 t + b_r)$$

- NTN is probably the most expressive embedding model to date
- Limitations: scalability issues

RESCAL vs. Distmult vs. NTN – Visualization

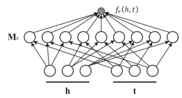

Figure: RESCAL

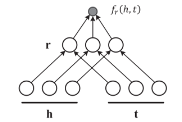

Figure: Distmult

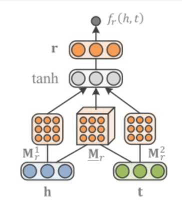

Figure: NTN

Incorpating additional information

Entity types and entity description

- Entity types
  - Consider IsA as an ordinary relation and the corresponding triples as ordinary training examples
  - Require entities in the same category to stay close in the embedding space
  - Take into account category hierarchy
- Entity description
  - Intialize the embedding of each entity using the embeddings of the words in its description
  - Jointly embed the entities and the words in their description

Training

Closed world assumption (CWA)

- The CWA assumes that all fact that are not observed in the knowledge graph are false
- Training under the closed world assumption only uses facts observed in the knowledge graph

DICE

- The CWA assumes that all fact that are not observed in the knowledge graph are false
- Training under the closed world assumption only uses facts observed in the knowledge graph
- For instance, training under the CWA can amount to solving the following optimization problem:

$$\min_{\Theta} \sum_{h,t \in \mathbb{E}, r \in \mathbb{R}} (y_{hrt} - f_r(h,t))^2$$

- Limitations: Cannot hold for incomplete knowledge graphs

Training
Open world assumption (OWA)

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 $\sum_{\tau^+ \in \mathbb{B}^+, \tau^- \in \mathbb{B}^-} \max(0, \gamma - f_r(h, t) + f_{r'}(h', t'))$
  Handle additional constraints or regularization terms
end loop

Output: Entity and relation embeddings

- Link prediction
- Triple classification
- Entity classification
- Entity resolution
- Relation extraction
- Question answering

Thanks

Thank you for listening