Neural Link Prediction for Multi-Modal Knowledge Graphs

Mathias Niepert and Alberto Garcia-Duran

NEC Labs Europe
Heidelberg
Outline

- Quick Reminder
- Failing with Latent and Relational Models
- Simple Link Prediction in KGs:
  - Graph Structure and Numerical Information
  - Visual Information
- Simple Link Prediction in Temporal KGs
- Remarks
Outline

Quick Reminder

Failing with Latent and Relational Models

Simple Link Prediction in KGs:
- Graph Structure and Numerical Information
- Visual Information

Simple Link Prediction in Temporal KGs

Remarks
## Tensor Factorization problem
- One adjacency matrix per relationship
- This is not new!

## Latent Models
- Scoring function operating on latent space
- At a high level, relationships can be seen as operations/transformations operating on the entities
- Parameter sharing between head and tail arguments

## Relational Models
- Extraction of relational features (e.g. via rule miners such as AMIE+)
- Scoring function operating on relational features

## Evaluation Metrics
- Queries of the type \((h, r, ?)\) or \((?, r, t)\)
- MRR is the most informative evaluation metric
Outline

Quick Reminder

Failing with Latent and Relational Models

Simple Link Prediction in KGs:
- Graph Structure and Numerical Information
- Visual Information

Simple Link Prediction in Temporal KGs

Remarks
“Compositionality” [Guu et al., 2016]

- They perform random walks in the KG and recursively apply the same transformation

Perform two random walks in the graph:

(John, has-ancestor, lives-in, LA)
(John, born-in, LA)
Failing with Latent and Relational Models

Latent Models

Apply the same transformation recursively (e.g. TransE):

If this happens this very often, then:

This amounts to learn the horn rule:

\[(h, \text{has-ancestor}, x) \land (x, \text{lives-in}, t) \rightarrow (h, \text{born-in}, t)\]
Failing with Latent and Relational Models

Latent Models

Let’s perform two more random walks:

\[ \text{John} + \text{ancestor} + \text{ancestor} \approx \text{Peter} \]

\[ \text{John} + \text{ancestor} \approx \text{Peter} \]

This happens this very often, then:

\[ \text{ancestor} + \text{ancestor} \approx \text{ancestor} \]

Only possible if the embedding for ancestor is 0...

... but this would collapse the embeddings of John, Mary and Peter to the same point. Not good...
For relational models to learn the predictive power of paths is easy.

- As long as we have enough examples of each path

But the number of possible paths grows exponentially with the number of relationships and length of the path:

- has-ancestor ^ has-ancestor
- has-ancestor ^ has-ancestor ^ lives-in
- has-ancestor ^ lives-in
- (born-in)^{-1} ^ has-ancestor
- (born-in)^{-1} ^ has-ancestor ^ has-ancestor

Try this
We cannot mine all possible paths in a graph
- No paths, no party!

[Neelankatan et al., 2015; Gardner et al., 2015] used relational features other than paths
- Path-bigram features
- One-sided features

For medium/large KGs the space of possible relational features is huge
- AMIE fails to obtain rules whose bodies are of up to 2 atoms in the Decagon data set (20k entities and 1k relation types) [Zitnik et al., 2018]
Failing with Latent and Relational Models
Take-Home Message

- Latent methods learn (at least) entity type information
- Latent methods very helpful for sparse KGs
- Latent models fail at learning very simple horn rules
- Dichotomy of relational features: either work perfectly or fail completely (random)
- Relational features outperform embedding methods on KBs with dense relational structure

<table>
<thead>
<tr>
<th>Query</th>
<th>Correct entity</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>nationality(?, US)</td>
<td>W. H. Macy</td>
<td>2</td>
</tr>
<tr>
<td>born_here(HK, ?)</td>
<td>W. Chau-sang</td>
<td>5</td>
</tr>
<tr>
<td>contains(?, Curtis-Inst)</td>
<td>USA</td>
<td>32</td>
</tr>
<tr>
<td>children(?, H. Roshan)</td>
<td>R. Roshan</td>
<td>26</td>
</tr>
</tbody>
</table>
Can we use latent models in conjunction with "simple" relational features?

Acknowledgement: There are a number of recent approaches [Rocktaschel et al., 2015; Guo et al., 2016; Minervini et al., 2017] that combine relational and latent representations by explicitly incorporating known logical rules into the embedding learning formulation.

[Guu et al., 2016] implicitly learns these logical rules! But we have learned that latent models have problem to learn a simple rule like:

\[ \text{has-ancestor} \land \text{has-ancestor} \rightarrow \text{has-ancestor} \]

In general, they have problems to learn rules wherein the relationship of the head of the rule also appears in the body:

\[ \text{rel}_1 \land \text{rel}_2 \rightarrow \text{rel}_1 \]
\[ \text{rel}_1 + \text{rel}_2 \neq \text{rel}_1 \text{ (TransE)} \]
\[ \text{rel}_1 \ast \text{rel}_2 \neq \text{rel}_1 \text{ (distMult, RESCAL)} \]
Outline

- Quick Reminder
- Failing with Latent and Relational Models
- Simple Link Prediction in KGs:
  - Graph Structure and Numerical Information
  - Visual Information
- Simple Link Prediction in Temporal KGs
- Remarks
Learn from all available information in the knowledge graph

Latent and Relational models only use graph structure information

[Wang et al., 2016; An et al., 2018] exploit textual information of entities and relationships

What about other data modalities?
Simple Link Prediction in KGs

- KG is given as a set of observed triples of the form \((h, r, t)\)

- We aim to combine an arbitrary number of feature types \(F\)

- **KBlrn** [Garcia-Duran et al., 2017a]: It is a product of experts approach wherein one expert is trained for each (relation type \(r\), feature type \(F\)) pair
  - We focus on latent, relational and numerical features
  - Generally, we may have more than an expert for the same feature type
**Product of Experts** [Hinton, 2000]:

\[
p(d|\theta_1 \ldots \theta_n) = \frac{\prod_m p_m(d|\theta_m)}{\sum_c \prod_m p_m(c|\theta_m)}
\]

where \( c \) indexes all possible vectors in the data space.

**From** [Hinton, 2000]:

“... so long as \( p_m \) is positive it does not need to be a probability at all ...”
We pick a latent model (e.g. distMult)

\[ d = (h, r, t) \]

and force its output to be positive

\[ f_{(r,L)}(d \mid \theta_{(r,L)}) = \exp((e_h \ast e_t) \cdot w^r) \]
Simple Link Prediction in KGs

KBlrn: Relational Expert

- Horn rules whose bodies have up to 2 atoms

![Diagram](https://via.placeholder.com/150)

- We use AMIE+ for the mining of closed horn rules

<table>
<thead>
<tr>
<th>Body of the Rule (Path)</th>
<th>Head of the Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\exists x (h, containedby, x) \land (t, locations_in_this_time_zone, x)$</td>
<td>$(h, time_zone, t)$</td>
</tr>
<tr>
<td>$\exists x (t, prequel, x) \land (x, character, h)$</td>
<td>$(h, character_in_film, t)$</td>
</tr>
<tr>
<td>$\exists x (x, cause_of_death, h) \land (x, cause_of_death, t)$</td>
<td>$(h, includes_causes_of_death, t)$</td>
</tr>
</tbody>
</table>

- “Bag-of-paths” for each relationship

- Relational expert: $d = (h, r, t)$

$$f_{(r,R)}(d \mid \theta_{(r,R)}) = \exp \left( r_{(h,t)} \cdot w^r_{\text{rel}} \right)$$
Simple Link Prediction in KGs

**KBlrn: Latent and Relational Expert**

**Product of Experts:**

\[
p(d | \theta_1 \ldots \theta_n) = \frac{\Pi_m p_m(d | \theta_m)}{\sum_c \Pi_m p_m(c | \theta_m)}
\]

**KBlr:**

\[
p(d \mid \theta_1, \ldots, \theta_n) = \frac{\prod_{F \in \{R,L\}} f(r,F)(d \mid \theta_{(r,F)})}{\sum_c \prod_{F \in \{R,L\}} f(r,F)(c \mid \theta_{(r,F)})}
\]
In practice this amounts to...

1. Sample N negative triples per positive triple
2. Compute scores of embedding and relational model for each
3. Sum scores and apply softmax function
4. Apply categorical cross-entropy loss

\[ \sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \]

\[ \text{softmax}(\begin{array}{c}
\text{...}
\end{array}) \leftarrow \mathcal{L} \]
Simple Link Prediction in KGs
KBln: Performance

Competitive with more complex KB completion models
FB15k, FB15k-237, FB122, WN18

**Metrics:**
- **MR:** Mean rank of correct triple
- **MRR:** Mean reciprocal rank
- **Hits@1:** Percentage of correct triples ranked 1
- **Hits@10:** Percentage of correct triples ranked in the top 10

<table>
<thead>
<tr>
<th></th>
<th>FB15k-237</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MR</td>
<td>MRR</td>
<td>Hits@1</td>
<td>Hits@10</td>
</tr>
<tr>
<td><strong>TRANSE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISTMult[27]</td>
<td>-</td>
<td>19.1</td>
<td>10.6</td>
<td>37.6</td>
</tr>
<tr>
<td>COMPLEX[27]</td>
<td>-</td>
<td>20.1</td>
<td>11.2</td>
<td>38.8</td>
</tr>
<tr>
<td>NODE+LINKFeat[29]</td>
<td>-</td>
<td>23.7</td>
<td>-</td>
<td>36.0</td>
</tr>
<tr>
<td>R-GCN+[27]</td>
<td>-</td>
<td>24.8</td>
<td>15.3</td>
<td>41.7</td>
</tr>
<tr>
<td>CONVE[4]</td>
<td>330</td>
<td>30.1</td>
<td>22.0</td>
<td>45.8</td>
</tr>
<tr>
<td><strong>KBL</strong></td>
<td>231</td>
<td>30.1</td>
<td>21.4</td>
<td>47.5</td>
</tr>
<tr>
<td><strong>KBR</strong></td>
<td>2518</td>
<td>18.0</td>
<td>12.8</td>
<td>28.5</td>
</tr>
<tr>
<td><strong>KBLR</strong></td>
<td><strong>231</strong></td>
<td><strong>30.6</strong></td>
<td><strong>22.0</strong></td>
<td><strong>48.2</strong></td>
</tr>
</tbody>
</table>
Numerical information is very common in knowledge bases (DBpedia, Freebase, YAGO, etc.)

Examples: geocoordinates, elevation, area, birth year, ...

Tokyo

<table>
<thead>
<tr>
<th>Area</th>
<th>2,187.66 km² (844.66 sq mi)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Metropolis</td>
<td>2,187.66 km² (844.66 sq mi)</td>
</tr>
<tr>
<td>• Metro</td>
<td>13,572 km² (5,240 sq mi)</td>
</tr>
<tr>
<td>Area rank</td>
<td>45th</td>
</tr>
</tbody>
</table>

| Elevation       | 40 m (130 ft)                 |

<table>
<thead>
<tr>
<th>Population (July 31, 2016)[3]</th>
<th>13,617,445</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Metropolis</td>
<td>13,617,445</td>
</tr>
<tr>
<td>• Density</td>
<td>6,224.66/km² (16,121.8/sq mi)</td>
</tr>
<tr>
<td>• Metro</td>
<td>37,800,000</td>
</tr>
<tr>
<td>• Metro density</td>
<td>2,662/km² (6,890/sq mi)</td>
</tr>
<tr>
<td>• 23 Wards</td>
<td>8,967,665</td>
</tr>
<tr>
<td></td>
<td>(2015 per prefectural government)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Demonym(s)</th>
<th>GetMethod, GetMethod, GetMethod</th>
</tr>
</thead>
<tbody>
<tr>
<td>• (Method)</td>
<td>GetMethod, GetMethod, GetMethod</td>
</tr>
<tr>
<td>• (Method)</td>
<td>GetMethod, GetMethod, GetMethod</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GDP (Nominal; 2014)[4][5]</th>
<th>$2 trillion</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Total</td>
<td>$2 trillion</td>
</tr>
<tr>
<td>• Per capita</td>
<td>$70,000</td>
</tr>
</tbody>
</table>
Simple Link Prediction in KGs

How to learn from them?
- Concatenate everything into a vector and pass it to whatever NN 😊

Observation: even though numerical features are not distributed according to a normal distribution, usually the difference between the head and tail arguments is
Simple Link Prediction in KGs

KBlrn: Numerical Expert

We use the difference values $n_{(h,t)}$ and the fact that they often follow a normal distribution. Why?

$$n_{(h,t)} = n_h - n_t \quad \quad N(c, \sigma) = n_h - n_t \quad \quad n_t = n_h + N(c, \sigma)$$

Numerical Expert:

$$f_{(x,N)}(d | \theta_{(x,N)}) = \exp \left( \phi \left( n_{(h,t)} \right) \cdot w_{num}^r \right)$$

where

$$\phi \left( n^{(i)}_{(h,t)} \right) = \exp \left( \frac{-||n^{(i)}_{(h,t)} - c_i||_2^2}{\sigma_i^2} \right)$$

Learning from the residual of the underlying linear regression model!

The output of the RBF is a value between 0 and 1.
Simple extension of previous model

Difference of numerical features + RBF layer

All parameters of the model are learned end-to-end
Numerical features helpful for KB completion

**Metrics:**
- **MR:** Mean rank of correct triple
- **MRR:** Mean reciprocal rank
- **Hits@1:** Percentage of correct triples ranked 1
- **Hits@10:** Percentage of correct triples ranked in the top 10

<table>
<thead>
<tr>
<th></th>
<th>FB15k-num</th>
<th></th>
<th>FB15k-237-num</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MR</td>
<td>MRR</td>
<td>Hits@1</td>
<td>Hits@10</td>
</tr>
<tr>
<td>TRANSER</td>
<td>39</td>
<td>72.6</td>
<td>62.1</td>
<td>89.7</td>
</tr>
<tr>
<td>DISTMULT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>without numerical features</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KBLR</td>
<td>39</td>
<td>72.6</td>
<td>62.1</td>
<td>89.7</td>
</tr>
<tr>
<td>KBRR</td>
<td>399</td>
<td>84.7</td>
<td><strong>81.6</strong></td>
<td>90.1</td>
</tr>
<tr>
<td>KBLRN</td>
<td>28</td>
<td>85.3</td>
<td>80.3</td>
<td>92.4</td>
</tr>
<tr>
<td>with numerical features</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KBLNR</td>
<td>32</td>
<td>73.6</td>
<td>63.0</td>
<td>90.7</td>
</tr>
<tr>
<td>KBRNR</td>
<td>68</td>
<td>84.0</td>
<td>80.6</td>
<td>90.0</td>
</tr>
<tr>
<td>KBLRN</td>
<td><strong>25</strong></td>
<td><strong>85.9</strong></td>
<td><strong>81.0</strong></td>
<td><strong>92.9</strong></td>
</tr>
</tbody>
</table>
Let’s have a deeper look at the relational expert:

What happens if we use a function other than a RBF:

<table>
<thead>
<tr>
<th>φ</th>
<th>MRR</th>
<th>Hits@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>sign</td>
<td>29.7</td>
<td>50.1</td>
</tr>
<tr>
<td>RBF</td>
<td>31.4</td>
<td>52.3</td>
</tr>
</tbody>
</table>
Outline

- Quick Reminder
- Failing with Latent and Relational Models

**Simple Link Prediction in KGs:**
- Graph Structure and Numerical Information
- **Visual Information**

- Simple Link Prediction in Temporal KGs

- Remarks
What about a KG of visual data?

**VisualGenome** [Krishna et al., 2015]
Simple Link Prediction in KGs

A KG where relationships hold between pairs of images?

We crawled images from search Engines using FB as a blueprint.
Simple Link Prediction in KGs

Query Types

1. Given two unseen images, predict the relationship

Given two unseen images, predict the relationship.

H

locatedIn

T

Tokyo

Japan

Gotoh Museum

Murasaki Shikibu

2. Given an unseen image and a relationship, retrieve related images

Given an unseen image and a relationship, retrieve related images.

Input Image Rank

Image

Relationship

hasArtAbout

Image Rank

1

2

3

....

H

locatedIn

T

locatedIn

capitalOf

locatedIn

Tokyo

Japan

Sensō-ji
**Simple Link Prediction in KGs**

**Query Types**

**Zero-Shot**: predicting relationships for new entities
Simple Link Prediction in KGs
Query Types

**Zero-Shot**: predicting relationships for new entities

- Gotoh Museum
- Murasaki Shikibu
- Sensō-ji
- Tokyo
- Japan
- (Mushashi)
**Zero-Shot**: predicting relationships for new entities

Simple Link Prediction in KGs

Query Types

- **Zero-Shot**: predicting relationships for new entities

![Diagram](image.png)
Simple Link Prediction in KGs

Query Types

**Zero-Shot**: predicting relationships for new entities

3. Unseen image -> Unseen image

![Diagram of Zero-Shot with two images and relationship types: artOf, locatedIn, bornIn, liveIn.]

4. Unseen image -> Seen images

![Diagram of Zero-Shot with two images and relationship types: artOf, locatedIn, bornIn, liveIn.]

- 0.4 – artOf
- 0.3 – locatedIn
- 0.1 – bornIn
- 0.1 – liveIn

Neural Link Prediction for Multi-Modal Knowledge Graphs
Simple Link Prediction in KGs

Image Graph [Onoro-Rubio et al., 2017]
- Acknowledgement: [Lonij et al., 2017]
- Function that learns latent representations from images

- Standard Scoring Functions
Given a pair of **unseen images** for which we do not know their KG entities, determine the **relations** between these underlying entities.

Performance Results:

<table>
<thead>
<tr>
<th>Model</th>
<th>Median</th>
<th>Hits@1</th>
<th>Hits@10</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>35</td>
<td>3.7</td>
<td>26.5</td>
<td>0.104</td>
</tr>
<tr>
<td>DIFF</td>
<td>11</td>
<td>21.1</td>
<td>50.0</td>
<td>0.307</td>
</tr>
<tr>
<td>MULT</td>
<td>8</td>
<td>15.5</td>
<td>54.3</td>
<td>0.282</td>
</tr>
<tr>
<td>CAT</td>
<td>6</td>
<td>26.7</td>
<td>61.0</td>
<td>0.378</td>
</tr>
<tr>
<td>DIFF+1HL</td>
<td>8</td>
<td>22.6</td>
<td>55.7</td>
<td>0.333</td>
</tr>
<tr>
<td>MULT+1HL</td>
<td>9</td>
<td>14.8</td>
<td>53.4</td>
<td>0.273</td>
</tr>
<tr>
<td>CAT+1HL</td>
<td>6</td>
<td>25.3</td>
<td>60.0</td>
<td>0.365</td>
</tr>
</tbody>
</table>

**Metrics:**
- **Median** – Median rank of the correct answer
- **Hits@1** – Percentage of correct answers ranked top 1
- **Hits@10** – Percentage of correct answers ranked in the top 10
- **MRR** – Mean reciprocal rank

+ 1HL (additional hidden layer)

Composition function $op$: difference (DIFF), multiplication (MULT), concatenation (CAT)
Simple Link Prediction in KGs
ImageGraph: Qualitative Results

**Relationship prediction**

<table>
<thead>
<tr>
<th>Input</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td>0.207025 film job</td>
</tr>
<tr>
<td><img src="image2.png" alt="Image" /></td>
<td>0.0438004 music_genre</td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td>0.0283168 people_with_this_profession</td>
</tr>
<tr>
<td><img src="image4.png" alt="Image" /></td>
<td>0.0272469 educational_degree</td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td>0.0248632 nominated_for</td>
</tr>
</tbody>
</table>

**Image retrieval**

- Paul Winfield
- Actor
- Pitcher
- Philadelphia Phillies

**Zero-Shot link prediction**

- ![Image](image6.png)
- ![Image](image7.png)
- ![Image](image8.png)
- ![Image](image9.png)
- ![Image](image10.png)

Relation Rank:
- Instrument played
- Tack contribution
- Musical group members

Relation Rank:
- Starring actor
- Award nominated work
- Character

Relation Rank:
- Film release region
- Featured film locations
- Place of birth

Relation Rank:
- BAFTA Award
- Award nominated work
- Award won
- Award nominee
Advantages & Disadvantages of:

- Latent models
- Relational models

KBlrn:

- A learning framework that combines multiple experts
- Reasoning with numerical information

Image Graph:

- A KG enriched with visual information
- Novel visual query types
- Zero-shot learning
5/10 minutes break
Outline

Quick Reminder

Failing with Latent and Relational Models

Simple Link Prediction in KGs:
- Graph Structure and Numerical Information
- Visual Information

Simple Link Prediction in Temporal KGs

Remarks
Research on link prediction has mainly focused on static KGs. What about time information?

Freebase, Yago, DBpedia... They all have time information in the original dumps.

Example of Temporal KG

```plaintext
Jamie_Lawrence playsFor Sunderland_A.F.C. occursUntil "1994-##-##"
Jeph_Loeb created Lost_(TV_series)
Bradley_Johnson playsFor Leeds_United_F.C. occursSince "2008-##-##"
Bradley_Johnson playsFor Leeds_United_F.C. occursUntil "2011-##-##"
David_Hibbert playsFor Shrewsbury_Town_F.C. occursSince "2007-##-##"
David_Hibbert playsFor Shrewsbury_Town_F.C. occursUntil "2010-##-##"
George_Eastham_Sr. isAffiliatedTo York_City_F.C.
Marcus_Gayle playsFor Rangers_F.C. occursSince "2001-##-##"
Iván_Campo playsFor AEK_Larnaca_F.C.
```
Simple Link Prediction in Temporal KGs

Multiple extremely sparse tensors

Not too much work...
- Temporal logic [van Benthem, 1995]
- Latent models [Jiang et al., 2016; Trivedi et al., 2017; Leblay et al., 2018]

Challenges:
- Time information is very sparse
- Heterogeneity of time information
- Point-in-Time / Intervals of time
(Not really) Off-topic: Language Modeling

**RNN Architecture**
- One representation per word
- Limited vocabulary

**Char - RNN Architecture**
- One representation per character
- Unlimited vocabulary
Simple Link Prediction in Temporal KGs

- [Garcia-Duran et al., 2018] Incorporate time information into standard embedding approaches for link prediction like

\[
\text{TRANSE}: \quad f(s, p, o) = ||e_s + e_p - e_o||_2 \\
\text{DISTMULT}: \quad f(s, p, o) = (e_s \circ e_o) e_p^T.
\]

**Trick:**

<table>
<thead>
<tr>
<th>Fact</th>
<th>Predicate Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>(BO, country, US)</td>
<td>[country]</td>
</tr>
<tr>
<td>(BO, bornIn, US, 1961)</td>
<td>[born, 1y, 9y, 6y, 1y]</td>
</tr>
<tr>
<td>(BO, president, US, since, 2009-01)</td>
<td>[president, since, 2y, 0y, 0y, 9y, 01m]</td>
</tr>
</tbody>
</table>

**Architecture:**

\[
\text{TA-TRANSE}: \quad f(s, p_{seq}, o) = ||e_s + e_{p_{seq}} - e_o||_2 \\
\text{TA-DISTMULT}: \quad f(s, p_{seq}, o) = (e_s \circ e_o) e_{p_{seq}}^T.
\]
Simple Link Prediction in Temporal KGs

Temporal KGs

<table>
<thead>
<tr>
<th>Data set</th>
<th>YAGO15K</th>
<th>ICEWS ’14</th>
<th>ICEWS 05-15</th>
<th>WIKIDATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entities</td>
<td>15,403</td>
<td>6,869</td>
<td>10,094</td>
<td>11,134</td>
</tr>
<tr>
<td>Relationships</td>
<td>34</td>
<td>230</td>
<td>251</td>
<td>95</td>
</tr>
<tr>
<td>#Facts</td>
<td>138,056</td>
<td>96,730</td>
<td>461,329</td>
<td>150,079</td>
</tr>
<tr>
<td>#Distinct TS</td>
<td>198</td>
<td>365</td>
<td>4,017</td>
<td>328</td>
</tr>
</tbody>
</table>

Training
- [110,441] [29,381]
- [78,826] [368,962]
- [121,422]

Validation
- [13,815] [3,635]
- [8,941] [46,275]
- [14,374]

Test
- [13,800] [3,685]
- [8,963] [46,092]
- [14,283]

Queries
- (s, p, ?) / (s, p, ?, date) / (s, p, ?, temporal_modifer, date)
- (?, p, o) / (?, p, o, date) / (?, p, o, temporal_modifer, date)

Standard Evaluation Metrics
- MRR, MR, Hits@1, Hits@10
Simple Link Prediction in Temporal KGs

<table>
<thead>
<tr>
<th></th>
<th>YAGO15k</th>
<th></th>
<th>WIKIDATA</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR</td>
<td>MR</td>
<td>Hits@10</td>
<td>Hits@1</td>
</tr>
<tr>
<td>TTransE</td>
<td>32.1</td>
<td>578</td>
<td>51.0</td>
<td>23.0</td>
</tr>
<tr>
<td>TransE</td>
<td>29.6</td>
<td>614</td>
<td>46.8</td>
<td>22.8</td>
</tr>
<tr>
<td>DISTMULT</td>
<td>27.5</td>
<td>578</td>
<td>43.8</td>
<td>21.5</td>
</tr>
<tr>
<td>TA-TRANSE</td>
<td>32.1</td>
<td>564</td>
<td>51.2</td>
<td>23.1</td>
</tr>
<tr>
<td>TA-DISTMULT</td>
<td>29.1</td>
<td>551</td>
<td>47.6</td>
<td>21.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>ICEWS 2014</th>
<th></th>
<th>ICEWS 2005-15</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR</td>
<td>MR</td>
<td>Hits@10</td>
<td>Hits@1</td>
</tr>
<tr>
<td>TTransE</td>
<td>25.5</td>
<td>148</td>
<td>60.1</td>
<td>7.4</td>
</tr>
<tr>
<td>TransE</td>
<td>28.0</td>
<td>122</td>
<td>63.7</td>
<td>9.4</td>
</tr>
<tr>
<td>DISTMULT</td>
<td>43.9</td>
<td>189</td>
<td>67.2</td>
<td>32.3</td>
</tr>
<tr>
<td>TA-TRANSE</td>
<td>27.5</td>
<td>128</td>
<td>62.5</td>
<td>9.5</td>
</tr>
<tr>
<td>TA-DISTMULT</td>
<td>47.7</td>
<td>276</td>
<td>68.6</td>
<td>36.3</td>
</tr>
</tbody>
</table>

[playsFor, since, temporal_tokens(date)]

t-SNE representations of a predicate sequence for different dates
All data sets are available at:

https://github.com/nle-ml/mmkb
Outline

Quick Reminder

Failing with Latent and Relational Models

Simple Link Prediction in KGs:
- Graph Structure and Numerical Information
- Visual Information

Simple Link Prediction in Temporal KGs

Remarks
**Remarks**

**Simple Link Prediction**

- A very important evaluation problem
  - Remember KGs are far from complete...
  - ... but they are used as ground truth for completeness evaluation
  - FB15k: (US, contains, ?)
    - Around 1600 entities answer that query
    - Less than 1000 of these correct completions are in the ground truth

- Advice: Run exps. in many data sets

- Importance of the loss & hyperparameter choices [Kadlec et al., 2017]

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
<th>MRR</th>
<th>确</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>DistMult (orig) (Yang et al., 2015)</td>
<td>-</td>
<td>94.2</td>
<td>0.83</td>
<td>57.7</td>
<td>0.35</td>
<td></td>
</tr>
<tr>
<td>DistMult (Toutanova and Chen, 2015)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>79.7</td>
<td>0.555</td>
<td></td>
</tr>
<tr>
<td>DistMult (Trouillon et al., 2017)</td>
<td>-</td>
<td>93.6</td>
<td>0.822</td>
<td>82.4</td>
<td>0.654</td>
<td></td>
</tr>
<tr>
<td>Single DistMult (this work)</td>
<td>655</td>
<td>94.6</td>
<td>0.797</td>
<td>42.2</td>
<td>89.3</td>
<td>0.798</td>
</tr>
</tbody>
</table>
Remarks
Simple Link Prediction

You can learn a lot!
• Experiments are relatively fast
• Implementation of these methods is easy

```python
# distMult model
e1 = Input(shape=(num_negative + 1,), name="e1")
e2 = Input(shape=(1,), name="e2")
rel = Input(shape=(1,), name="rel")

# Embeddings
t embedding = Embedding(embEnt, emb_dim, name='ent_embeddings')
rel embedding = Embedding(embRel, emb_dim, name='rel_embeddings')

e1 emb = Dropout(drop)(ent embedding(e1))
e2 emb = Dropout(drop)(ent embedding(e2))
rel emb = Dropout(drop)(rel embedding(rel))

mult elmWise = Multiply(name='joint emb')([[rel emb, e2 emb]])
score = Dot(2, name='score')([mult elmWise, e1 emb])
score = Activation('softmax')(Reshape((num_negative + 1,), name='soft_out')(score))
adam = Adam()

model = Model(input=[e1, e2, rel], output=[score])
model.compile(loss='categorical_crossentropy', optimizer=adam)
```

Simple Link Prediction
• Don’t do it
• There are many other interesting related problems!
Link Prediction in Attributed KGs [Zitnik et al., 2018]
Remarks
Interesting Problems

Complex Queries [Hamilton et al., 2018]
Non-Binary Predicates
- ((Alberto, Mathias), met_at, Heidelberg)
- Reification: turn n-ary predicates to binary ones

Entity Linking
- Specific type of query (h, sameAs, ?) / (? , sameAs, t)

Hierarchical Embeddings in KGs?
- [Nickel et al., 2017] learn hierarchical embeddings for taxonomies

Get inspiration for other problems!
Remarks
Related Problems

Recommender Systems
- Sequential Recommendation [He et al., 2017]
Remarks

Related Problems

Recommender Systems

● Reviews available [Garcia-Duran et al., 2017b]
Remarks
Related Problems

Recommender Systems

![Diagram of recommender systems with user embedding, review embedding, and product embedding.]

- User embedding + review embedding = product embedding
- Example:
  - User rating: 4.0
  - Review: "Really nice bike"
  - Avg embedding:
    - Really
    - nice
    - bike
  - Similarity check:
    - Bike rating: 3.2
    - Review: "Bad suspension but good price"
Remarks
Related Problems

Language Modeling [Ahn et al., 2016]

**Example.** Given a topic on Fred Rogers with topic description

\[ W = \text{"Rogers was born in Latrobe, Pennsylvania in 1928"} \]

and topic knowledge \( \mathcal{F} = \{a^{42}, a^{83}, a^0\} \) where

\[ a^{42} = (\text{Fred.Rogers, Place.of.Birth, Latrobe.Pennsylvania}) \]
\[ a^{83} = (\text{Fred.Rogers, Year.of.Birth, 1928}) \]
\[ a^0 = (\text{Fred.Rogers, Topic.Itself, Fred.Rogers}) \]
### Text Generation [Serban et al., 2016]

<table>
<thead>
<tr>
<th>Fact</th>
<th>Human</th>
<th>Baseline</th>
<th>MP Triples TransE++</th>
</tr>
</thead>
<tbody>
<tr>
<td>bayuvi dupki – contained by – europe</td>
<td>where is bayuvi dupki?</td>
<td>what state is the city of bayuvi dupki located in?</td>
<td>what continent is bayuvi dupki in?</td>
</tr>
<tr>
<td>illinois – contains – ludlow township</td>
<td>what is in illinois?</td>
<td>what is a tributary found in illinois?</td>
<td>what is the name of a place within illinois?</td>
</tr>
<tr>
<td>neo contra – publisher – konami</td>
<td>who published neo contra?</td>
<td>which company published the game neo contra?</td>
<td>who is the publisher for the computer videogame neo contra?</td>
</tr>
<tr>
<td>pop music – artists – nikki flores</td>
<td>what artist is known for pop music?</td>
<td>An example of pop music is what artist?</td>
<td>who’s an american singer that plays pop music?</td>
</tr>
</tbody>
</table>
Remarks

Learned the basic concepts of:
- Relational Models
- Latent Models
- Strengths & Weaknesses

How to learn from modalities other than the graph structure

How to learn with temporal information

Many interesting problems to address
- Simple Link Prediction is not one of them

People are using KGs in many other problems

Conclusions
References

[Ahn et al.,2017] A Neural Knowledge Language Model
[An et al.,2018] Accurate Text-Enhanced Knowledge Graph Representation Learning
[van Benthem et al.,1995] Temporal Logic
[Garcia-Duran et al.,2017a] KBlrn: End-to-End Learning of Knowledge Base Representations with Latent, Relational and Numerical Features
[Garcia-Duran et al.,2017b] TransRev: Modeling Reviews as Translations from Users to Items
[Garcia-Duran et al.,2018] Learning Sequence Encoders for Temporal Knowledge Graph Completion
[Gardner et al.,2015] Efficient and Expressive Knowledge Base Completion Using Subgraph Feature Extraction
[Guo et al.,2016] Jointly Embedding Knowledge Graphs and Logical Rules
[Guu et al.,2016] Traversing the Knowledge Graph in Vector Space.
[Hamilton et al.,2018] Querying Complex Networks in Vector Space
[He et al.,2017] Translation-Based Recommendation
[Jiang et al.,2016] Encoding Temporal Information for Time-Aware Link Prediction
[Kadlec et al.,2017] Knowledge Base Completion: Baselines Strikes Back
References

[Krishna et al., 2015] Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations

[Leblay et al., 2018] Deriving Time Validity in Knowledge Graph


[Minervini et al., 2017] Adversarial Sets for Regularized Neural Link Predictors

[Neelankatan et al., 2015] Compositional Vector Space Models for Knowledge Base Completion

[Nickel et al., 2017] Poincare Embeddings for Learning Hierarchical Representations

[Rocktaschel et al., 2015] Injecting Logical Background Knowledge into Embeddings for Relation Extraction

[Serban et al., 2016] Generating Factoid Questions with RNN


[Wang et al., 2016] Text-Enhanced Representation Learning for Knowledge Graph

[Zitnik et al., 2018] Modeling Polypharmacy Side Effects with Graph Convolutional Networks