Knowledge Graph Embeddings: From Theory to Practice

Luca Costabello
Accenture Labs
@lukostaz

Sumit Pai
Accenture Labs
@sumitppai

Nicholas McCarthy
Accenture Labs
@nickpmcc

Adrianna Janik
Accenture Labs
@adri_janik
Outline

**Theoretical Overview**
- Introduction
- Anatomy of a Knowledge Graph Embedding Model
- Evaluation Protocol and Metrics
- Advanced KGE Topics
- Open Research Questions

**Applications**

**Software Ecosystem**

**Hands-on Session**
Outline

Theoretical Overview

- Introduction
- Anatomy of a Knowledge Graph Embedding Model
- Evaluation Protocol and Metrics
- Advanced KGE Topics
- Open Research Questions

Applications

Software Ecosystem

Hands-on Sessions
Knowledge Graph

\[ \mathcal{G} = \{(s, p, o)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E} \]

\( \mathcal{E} \) : set of entities of \( \mathcal{G} \)  
\( \mathcal{R} \) : set of relations of \( \mathcal{G} \)

In-depth overview of Knowledge Graphs in [Hogan et al. 2020]
SOCIAL NETWORKS

[neo4j.com]

COLLABORATIVE WEB-BASED KNOWLEDGE BASES

[lod-cloud.net]

PROTEIN-PROTEIN INTERACTION NETWORKS

[ebi.ac.uk]
<table>
<thead>
<tr>
<th>Knowledge Graph</th>
<th>Statements</th>
<th>Entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>yago</td>
<td>120 M</td>
<td>10 M</td>
</tr>
<tr>
<td>Wikidata</td>
<td>610 M</td>
<td>51 M</td>
</tr>
<tr>
<td>DBpedia</td>
<td>1.3 B</td>
<td>6 M</td>
</tr>
<tr>
<td>GDELT</td>
<td>3.5 B</td>
<td>364 M</td>
</tr>
</tbody>
</table>
Knowledge Graphs & The Open World Assumption

- **Closed World Assumption (CWA):** absence of a fact means it is necessarily false.
- **Open World Assumption (OWA):** absence of a fact does not imply fact is false. We simply do not know.
Machine Learning on Knowledge Graphs/
Statistical Relational Learning

**LINK PREDICTION / TRIPLE CLASSIFICATION**
- Knowledge graph completion
- Content recommendation
- Question answering

**COLLECTIVE NODE CLASSIFICATION / LINK-BASED CLUSTERING**
- Customer segmentation

**ENTITY MATCHING**
- Duplicate detection
- Inventory items deduplication

Pic from [Nickel et al. 2016a]
Assigning a score proportional to the likelihood that an unseen triple is true.

**Link Prediction**
- Learning to rank problem
- Information retrieval metrics
- No ground truth negatives in test set required

**Triple Classification**
- Binary classification task
- Binary classification metrics
- Test set requires positives and ground truth negatives
Traditional Statistical Relational Learning

- **Logic Programming**: predict new links from facts and extracted rules.
- **Inductive Logic Programming (ILP)**: predict new links from rules extracted from correlated facts.
- **Rule Mining**: e.g. AMIE+: extracts Horn clauses based on their support in the KG.
- **Graphical Models**:
  - Conditional Random Fields (CRFs)
  - Probabilistic Relational Models
  - Relational Markov Networks
  - Relational Dependency Networks

**Limitations**

- Limited Scalability to KG size
- Limited modeling power
- Non-differentiable approaches

[Getoor & Taskar 2007]

[Gallaraga et al. 2015]
Introducing (Graph) Representation Learning

• Manual feature engineering on graphs is hard and time-consuming...

• ... what if instead we learn representations of nodes and edges?

Components that learn from data

Pic from [Goodfellow et al. 2016]
Can we re-use traditional deep learning tools?

- CNNs are designed for grids (e.g. images)
- RNNs/word2vec for sequences (e.g. text)

But graphs are more complex:
- No spatial locality
- No fixed-node ordering (graph isomorphism problem)
- Multimodal (concepts, text, numbers, timestamps, etc.)

We need ad-hoc models!

From AAAI-19 tutorial on Graph Representation Learning [Hamilton & Tang 2019]
Graph Representation Learning
Learning representations of nodes and edges

Node Representation/Graph Feature based Methods
PRA, LINE, DeepWalk, node2vec

Graph Neural Networks (GNNs)
GCNs, Graph Attention Networks

Knowledge Graph Embeddings (KGE)
TransE, DistMult, ComplEx, ConvE

Scope of this tutorial

For a complete overview of graph feature-based models and GNNs:
[Hamilton & Sun 2019]
[Hamilton 2020]
Knowledge Graph Embeddings (KGE)

Automatic, supervised learning of **embeddings**, i.e. projections of entities and relations into a continuous low-dimensional space.
From Nodes and Edges ...
... To Semantically Meaningful Vector Representations
**Some KGE models** in recent published literature:

- **TransE** (Bordes et al., 2013)
- **RESCAL** (Nickel et al., 2011)
- **DistMult** (Yang et al., 2014)
- **ComplEx** (Trouillon et al., 2016)
- **HolE** (Nickel et al., 2016)
- **ConvE** (Dettmers et al., 2017)
- **ComplEx-N3** (Lacroix et al., 2018)
- **RotatE** (Sun et al., 2019)
KGE Design Rationale: Capture KG Patterns

Symmetry

<Alice marriedTo Bob>

Asymmetry

<Alice childOf Jack>

Inversion

<Alice childOf Jack>

<Jack fatherOf Alice>

Composite

<Alice childOf Jack>

<Jack siblingOf Mary>

<Alice nieceOf Mary>

But also:
• Hierarchies
• Type constraints
• Transitivity
• Homophily
• Long-range dependencies
<table>
<thead>
<tr>
<th>Model</th>
<th>Symmetry</th>
<th>Antisymmetry</th>
<th>Inversion</th>
<th>Composition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>TransE</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TransX</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>DistMult</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>ComplEx</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>RotatE</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

[Sun et al. 2019]
Outline

Theoretical Overview 1h 30m

• Introduction
• Anatomy of a Knowledge Graph Embedding Model
• Evaluation Protocol and Metrics
• Advanced KGE Topics
• Open Research Questions

Applications 15 m

Software Ecosystem 15 m

Hands-on Sessions 1h 15m
At a Glance

**Optimizer**

**Lookup Layer**

**Scoring Layer** $f(s, p, o) \in \mathbb{R}$

**Loss Functions** $\mathcal{L}$

**Negatives Generation**

**Training**

**Downstream Tasks** (Link Prediction)

$f(s, p, o) \in \mathbb{R}$

**Inference**
Anatomy of a Knowledge Graph Embedding Model

- Knowledge Graph (KG) $\mathcal{G}$
- Scoring function for a triple $f(t)$
- Loss function $\mathcal{L}$
- Optimization algorithm
- Negatives generation strategy
Scoring function $f$

$f$ assigns a score to a triple $(s, p, o)$

High score = high changes for the triple to be a true fact.

Translation-based Scoring Functions

- **TransE**: Translating Embeddings  
  
  $f_{TransE} = -|| (e_s + r_p) - e_o ||_n$

  [Bordes et al. 2013]
Translation-based Scoring Functions

- **RotatE**: relations modelled as rotations in complex space $\mathbb{C}$: element-wise product between complex embeddings.

$$f_{\text{RotatE}} = -|e_s \circ r_p - e_o|_n$$

[Sun et al. 2019]
Factorization-based Scoring Functions

• **RESCAL**: low-rank factorization with tensor product

\[ f_{RESCAL} = e_s^T W_r e_o \]

[Nickel et al. 2011]

• **DistMult**: bilinear diagonal model. Dot product.

\[ f_{DistMult} = \langle r_p, e_s, e_o \rangle \]

[Yang et al. 2015]

• **ComplEx**: Complex Embeddings (Hermitian dot product): (i.e. extends DistMult with dot product in \( \mathbb{C} \))

\[ f_{ComplEx} = Re(\langle r_p, e_s, \overline{e_o} \rangle) \]

[Trouillon et al. 2016]
“Deeper” Scoring Functions

- **ConvE**: reshaping + convolution

  \[
  f_{\text{ConvE}} = \langle \sigma \left( \text{vec}(g([e_s; \vec{r}_p] * \Omega)) \right) W \rangle e_o
  \]

- **ConvKB**: convolutions and dot product

  \[
  f_{\text{ConvKB}} = \text{concat} \left( g \left( [e_s, r_p, e_o] * \Omega \right) \right) \cdot W
  \]

  Computationally expensive!

[Dettmers et al. 2017]

[Nguyen et al. 2018]
Other Recent Models

• HoLE [Nickel et al. 2016]
• SimplE [Kazemi et al. 2018]
• QuatE [Zhang et al. 2019]
• MurP [Balažević et al. 2019]
• ...

**Loss function** $\mathcal{L}$

**Pairwise Margin-Based Hinge Loss**

Pays a penalty if score of positive triple $< \text{score of synthetic negative}$ by a margin $\gamma$

$$\mathcal{L}(\Theta) = \sum_{t^+ \in G} \sum_{t^- \in \mathcal{C}} \max(0, [\gamma + f(t^-; \Theta) - f(t^+; \Theta)])$$

Score assigned to a synthetic negative

Score assigned to true triple

[Bordes et al. 2013]

**Negative Log-Likelihood / Cross Entropy**

$$\mathcal{L}(\Theta) = \sum_{t \in G \cup \mathcal{C}} \log(1 + \exp(-y f(t; \Theta)))$$

Label of the triple $t$

$y \in \{-1, 1\}$

[Trouillon et al. 2016]
Loss function $\mathcal{L}$

**Binary Cross-Entropy**

$$\mathcal{L} = -\frac{1}{N} \sum_{t \in G \cup \mathcal{C}} y \cdot \log(\sigma(f(t; \Theta))) + (1 - y) \cdot \log(1 - f(t; \Theta))$$

[Dettmers et al. 2017]

**Self-Adversarial**

$$\mathcal{L} = -\log \sigma(\gamma + f(t^+; \Theta)) - \sum_{t \in G} p(t^-; \Theta) \log \sigma(-f(t^-; \Theta) - \gamma)$$

[Weight for the negative sample $t^-$.] [Sun et al. 2019]

**Many more:** Multiclass Negative Log-likelihood, Absolute Margin, etc.
Regularizers
• L1, L2
• L3  [Lacroix et al. 2018]
• Dropout (ConvE)  [Dettmers et al. 2017]

Initialization
• Random (Uniform)
• Random (Normal)
• Glorot
Negatives Generation

Where do negative examples come from? (i.e. false facts)

**Local Closed World Assumption:** the KG is only *locally* complete

“Corrupted” versions of a triple as synthetic negatives:

\[
\mathcal{C} = \{(\hat{s}, p, o) | \hat{s} \in \mathcal{E}\} \cup \{(s, p, \hat{o}) | \hat{o} \in \mathcal{E}\}
\]

"corrupted subject" 

"corrupted" object

The predicate is unaltered

\[
f(s, p, o) \in \mathbb{R}
\]
**Synthetic Negatives: Example**

\[ \mathcal{E} = \{Mike, Liverpool, AcmeInc, George, LiverpoolFC\} \]

\[ \mathcal{R} = \{\text{bornIn, friendWith}\} \]

\[ t \in \mathcal{G} = (\text{Mike bornIn Liverpool}) \]

\[ C_t = \]

<table>
<thead>
<tr>
<th>Mike</th>
<th>bornIn</th>
<th>AcmeInc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>bornIn</td>
<td>LiverpoolFC</td>
</tr>
<tr>
<td>George</td>
<td>bornIn</td>
<td>Liverpool</td>
</tr>
<tr>
<td>AcmeInc</td>
<td>bornIn</td>
<td>Liverpool</td>
</tr>
</tbody>
</table>
Training with Synthetic Negatives

**Uniform sampling**: generate all possible synthetic negatives and sample n negatives for each positive t.

**Complete set**: no sampling. Use all possible synthetic negatives for each positive t. (mind scalability)

**1-n scoring**: batches of (s, p, *) or (*, p, o) labeled as positives (if included in training KG) or negatives (if not in training KG).

[Dettmers et al. 2017]
Training Procedure and Optimizer

Optimizer: learn optimal parameters (e.g. embeddings). Off-the-shelf SGD variants: (AdaGrad, Adam)

\[ \min_{\Theta} \mathcal{L}(\Theta) \]

Reciprocal Triples
Injection of reciprocal triples in training set.

<Alice childOf Jack>  
<Jack childOf^{-1} Alice>  

[Dettmers et al. 2017]  
[Lacroix et al. 2018]
Model Selection

• Grid search
  • Mind the size of the grid!
  • Early stopping

• Random search

• Quasi-random + Bayesian  [Ruffinelli et a. 2020]
Outline

Theoretical Overview

• Introduction
• Anatomy of a Knowledge Graph Embedding Model
• Evaluation Protocol and Metrics
• Advanced KGE Topics
• Open Research Questions

Applications 15 m

Software Ecosystem 15 m

Hands-on Sessions 1h 15m
Assigning a score proportional to the likelihood that an unseen triple is true.

**Link Prediction**
- Learning to rank problem
- Information retrieval metrics
- No ground truth negatives in test set required

**Triple Classification**
- Binary classification task
- Binary classification metrics
- Test set requires positives and ground truth negatives

**Learning–To-Rank problem:** How well are positive triples ranked against synthetic negatives built under the **Local Closed World Assumption.**
Evaluation Metrics

Mean Rank (MR)

\[ MR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} rank_{(s,p,o)}(i) \]

Mean Reciprocal Rank (MRR)

\[ MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_{(s,p,o)}(i)} \]

Hits@N

\[ Hits@N = \sum_{i=1}^{|Q|} 1 \text{ if } rank_{(s,p,o)}(i) \leq N \]
### Example

How unseen, test positive triples rank against **synthetic negatives**?

<table>
<thead>
<tr>
<th>s</th>
<th>p</th>
<th>o</th>
<th>score</th>
<th>rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>born_in</td>
<td>Leeds</td>
<td>0.789</td>
<td>1</td>
</tr>
<tr>
<td>Mike</td>
<td>born_in</td>
<td>Liverpool</td>
<td>0.753</td>
<td>2</td>
</tr>
<tr>
<td>Mike</td>
<td>born_in</td>
<td>Germany</td>
<td>0.695</td>
<td>3</td>
</tr>
<tr>
<td>George</td>
<td>born_in</td>
<td>Liverpool</td>
<td>0.456</td>
<td>4</td>
</tr>
<tr>
<td>Mike</td>
<td>born_in</td>
<td>George</td>
<td>0.234</td>
<td>5</td>
</tr>
</tbody>
</table>

Unseen positive triples (test set)

<table>
<thead>
<tr>
<th>s</th>
<th>p</th>
<th>o</th>
<th>score</th>
<th>rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>friend_with</td>
<td>George</td>
<td>0.901</td>
<td>1</td>
</tr>
<tr>
<td>Mike</td>
<td>friend_with</td>
<td>Jim</td>
<td>0.345</td>
<td>2</td>
</tr>
<tr>
<td>Acme</td>
<td>friend_with</td>
<td>George</td>
<td>0.293</td>
<td>3</td>
</tr>
<tr>
<td>Mike</td>
<td>friend_with</td>
<td>Liverpool</td>
<td>0.201</td>
<td>4</td>
</tr>
<tr>
<td>France</td>
<td>friend_with</td>
<td>George</td>
<td>0.156</td>
<td>5</td>
</tr>
</tbody>
</table>

\[
MR = 1.5
\]

\[
MRR = 0.75
\]

\[
Hits@1 = 0.5
\]

\[
Hits@3 = 1.0
\]
## Benchmark Datasets

<table>
<thead>
<tr>
<th></th>
<th>FB15K-237</th>
<th>WN18RR</th>
<th>YAGO3-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>272,115</td>
<td>86,835</td>
<td>1,079,040</td>
</tr>
<tr>
<td>Validation</td>
<td>17,535</td>
<td>3,034</td>
<td>5,000</td>
</tr>
<tr>
<td>Test</td>
<td>20,466</td>
<td>3,134</td>
<td>5,000</td>
</tr>
<tr>
<td>Entities</td>
<td>14,541</td>
<td>40,943</td>
<td>123,182</td>
</tr>
<tr>
<td>Relations</td>
<td>237</td>
<td>11</td>
<td>37</td>
</tr>
</tbody>
</table>
## Link Prediction: SOTA Results

<table>
<thead>
<tr>
<th></th>
<th>FB15K-237</th>
<th>WN18RR</th>
<th>YAGO3-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Literature Best</td>
<td>0.35*</td>
<td>0.48*</td>
<td>0.49*</td>
</tr>
<tr>
<td>TransE (AmpliGraph)</td>
<td>0.31</td>
<td>0.22</td>
<td>0.51</td>
</tr>
<tr>
<td>DistMult (AmpliGraph)</td>
<td>0.31</td>
<td>0.47</td>
<td>0.50</td>
</tr>
<tr>
<td>ComplEx (AmpliGraph)</td>
<td>0.32</td>
<td>0.51</td>
<td>0.49</td>
</tr>
<tr>
<td>HolE (AmpliGraph)</td>
<td>0.31</td>
<td>0.47</td>
<td>0.50</td>
</tr>
<tr>
<td>ConvE (AmpliGraph)</td>
<td>0.26</td>
<td>0.45</td>
<td>0.30</td>
</tr>
<tr>
<td>ConvE (1-N, AmpliGraph)</td>
<td>0.32</td>
<td>0.48</td>
<td>0.40</td>
</tr>
<tr>
<td>ConvKB (AmpliGraph)</td>
<td>0.23</td>
<td>0.39</td>
<td>0.30</td>
</tr>
</tbody>
</table>

**First**

<table>
<thead>
<tr>
<th>Method</th>
<th>FB15K-237 MRR</th>
<th>FB15K-237 Hits@10</th>
<th>WNRR MRR</th>
<th>WNRR Hits@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESCAL (Wang et al., 2019)</td>
<td>27.0</td>
<td>42.7</td>
<td>42.0</td>
<td>44.7</td>
</tr>
<tr>
<td>TransE (Nguyen et al., 2018)</td>
<td>29.4</td>
<td>46.5</td>
<td>22.6</td>
<td>50.1</td>
</tr>
<tr>
<td>DistMult (Dettmers et al., 2018)</td>
<td>24.1</td>
<td>41.9</td>
<td>43.0</td>
<td>49.0</td>
</tr>
<tr>
<td>ComplEx (Dettmers et al., 2018)</td>
<td>24.7</td>
<td>42.8</td>
<td>44.0</td>
<td>51.0</td>
</tr>
<tr>
<td>ConvE (Dettmers et al., 2018)</td>
<td>32.5</td>
<td>50.1</td>
<td>43.0</td>
<td>52.0</td>
</tr>
</tbody>
</table>

**Ours**

<table>
<thead>
<tr>
<th>Method</th>
<th>FB15K-237 MRR</th>
<th>FB15K-237 Hits@10</th>
<th>WNRR MRR</th>
<th>WNRR Hits@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESCAL</td>
<td>35.7</td>
<td>54.1</td>
<td>46.7</td>
<td>51.7</td>
</tr>
<tr>
<td>TransE</td>
<td>31.3</td>
<td>49.7</td>
<td>22.8</td>
<td>52.0</td>
</tr>
<tr>
<td>DistMult</td>
<td>34.3</td>
<td>53.1</td>
<td>45.2</td>
<td>53.1</td>
</tr>
<tr>
<td>ComplEx</td>
<td>34.8</td>
<td>53.6</td>
<td>47.5</td>
<td>54.7</td>
</tr>
<tr>
<td>ConvE</td>
<td>33.9</td>
<td>52.1</td>
<td>44.2</td>
<td>50.4</td>
</tr>
</tbody>
</table>

**Recent**

<table>
<thead>
<tr>
<th>Method</th>
<th>FB15K-237 MRR</th>
<th>FB15K-237 Hits@10</th>
<th>WNRR MRR</th>
<th>WNRR Hits@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>TuckER (Balazevic et al., 2019)</td>
<td>35.8</td>
<td>54.4</td>
<td>47.0</td>
<td>52.6</td>
</tr>
<tr>
<td>RotatE (Sun et al., 2019a)</td>
<td>33.8</td>
<td>53.3</td>
<td><strong>47.6</strong></td>
<td><strong>57.1</strong></td>
</tr>
<tr>
<td>SACN (Shang et al., 2019)</td>
<td>35.0</td>
<td>54.0</td>
<td>47.0</td>
<td>54.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>FB15K-237 MRR</th>
<th>FB15K-237 Hits@10</th>
<th>WNRR MRR</th>
<th>WNRR Hits@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DistMult (Salehi et al., 2018)</td>
<td>35.7</td>
<td>54.8</td>
<td>45.5</td>
<td>54.4</td>
</tr>
<tr>
<td>ComplEx-N3 (Lacroix et al., 2018)</td>
<td><strong>37.0</strong></td>
<td><strong>56.0</strong></td>
<td><strong>49.0</strong></td>
<td><strong>58.0</strong></td>
</tr>
</tbody>
</table>

[https://github.com/Accenture/AmpliGraph](https://github.com/Accenture/AmpliGraph)

[Ruffinelli et a. 2020]
Comparing SOTA Results is Tricky

- Different training strategies (e.g. synthetic negatives)
- Reciprocal relations in training set?
- Unfair or suboptimal hyperparameters selection
- Evaluation protocol: how to behave with tie ranks?
- Ablation studies!

Read discussion in [Ruffinelli et al 2020]
Outline

Theoretical Overview 1h 30m
- Introduction
- Anatomy of a Knowledge Graph Embedding Model
- Evaluation Protocol and Metrics
- Advanced KGE Topics
- Open Research Questions

Applications 15 m

Software Ecosystem 15 m

Hands-on Sessions 1h 15m
Calibration

Probabilities Generated by KGE models are Uncalibrated!

$ f_m(\cdot) = 0.75 $ confidence $ \Rightarrow $ model should be correct 75% of the times.

- **Mistrust** in model discoveries
- **Poor Interpretability** in high-stakes scenarios (i.e. drug-target discovery)

How can we calibrate KGE models? How to do so without ground truth negatives?

[Tabacof & Costabello 2020]
Calibrating With **Ground Truth Negatives**  
*(Available w/ Triple Classification Datasets)*

**Trained KG Embedding Model**  
\[ f(s, p, o) \in \mathbb{R}^k \]

**Calibration**  
Platt Scaling/Isotonic Regression

**Calibrated Probabilities**

<table>
<thead>
<tr>
<th></th>
<th>Calibrated Probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
</tr>
<tr>
<td></td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td>0.87</td>
</tr>
</tbody>
</table>

**Positive Base Rate**  
(Automatically Inferred)

**Calibration dataset**  
(Ground truth positives + negatives)

**Inference-time scores (logits)**

<table>
<thead>
<tr>
<th></th>
<th>Inference-time scores (logits)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-0.25</td>
</tr>
<tr>
<td></td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>-0.13</td>
</tr>
<tr>
<td></td>
<td>0.21</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

[Tabacof & Costabello 2020]
Calibrating With **Synthetic Negatives**

(No Ground Truth Negatives Available)

[Tabacof & Costabello 2020]
Calibration is Effective

[Tabacof & Costabello 2020]

- All calibration techniques work **considerably better** than uncalibrated settings
- More **trustworthy and interpretable** predictions
Many real-world graphs include multi-modal attributes.

Dublin (/ˈdɒblɪn/, Irish: Baile Átha Cliath [ˈblʲathəˈklʲiːat̪ˠ]) is the capital and largest city of Ireland. Dublin is in the province of Leinster on Ireland's east coast, at the mouth of the River Liffey. The city has an urban area population of 1,345,402. The population of the Greater Dublin Area, as of 2016, was 1,904,806 people. Founded as a Viking settlement, the Kingdom of Dublin became Ireland's principal city following the Norman invasion. The city expanded rapidly from the 17th century and was briefly the second largest city [...]

Multimodal Knowledge Graphs
Multimodal Knowledge Graph Embeddings

- **KBLRN** [Garcia-duran et al. 2017]
- **LiteralE** [Kristiadi et al. 2018]
- **MKBE** [Pezeshkpour et al. 2018]

[Gesese et al. 2019] surveys recent literature
Temporal Knowledge Graphs

Many real-world graphs represent timestamped concepts.

Table from [Lacroix et al. 2020]
**Time Awareness: Temporal KGE models**

**TNTComplEx**
- Embeddings for each timestamp
- Order 4 tensor decomposition problem
- ComplEx as decomposition method

<table>
<thead>
<tr>
<th>Model</th>
<th>ICEWS14</th>
<th>ICEWS15-05</th>
<th>Yago15k</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA</td>
<td>0.48</td>
<td>0.47</td>
<td>0.32</td>
</tr>
<tr>
<td>DE-SimplE</td>
<td>0.53</td>
<td>0.51</td>
<td>-</td>
</tr>
<tr>
<td>ComplEx</td>
<td>0.47 (0.47)</td>
<td>0.49 (0.49)</td>
<td>0.35 (0.36)</td>
</tr>
<tr>
<td>TComplEx</td>
<td>0.56 (0.61)</td>
<td>0.58 (0.66)</td>
<td>0.35 (0.36)</td>
</tr>
<tr>
<td>TNTComplEx</td>
<td>0.56 (0.62)</td>
<td>0.60 (0.67)</td>
<td>0.35 (0.37)</td>
</tr>
</tbody>
</table>

[Lacroix et al. 2020]
Uncertain Knowledge Graphs

Automatic KG generation may lead to uncertain facts.

UKGE [Chen et al. 2019]

• Jointly training of KGE model + probabilistic soft logic to predict likelihood of unseen triples
• Logical rules are required as additional input
Robustness
KGE suffer from adversarial modifications

Link Prediction

(a) KG, with the target prediction
(b) After removing a fact
(c) After adding a fact

Node Classification

(a) Original
(b) Adversarial perturbations
(c) Adversarial attack
(d) Adversarial robustness
Robustness

Zhang et al. 2019
Generates input perturbations from the latent space by scoring all possible perturbations

CRIAGE [Pezeshkpour et al. 2019]
Encoder-decoder based inverter neural network

(a) Direct Adding Attack on WN18 Dataset against TransE
(b) Direct Deleting Attack on WN18 Dataset against TransE
KGE & Neuro-Symbolic Reasoning

Background knowledge injection with Soft Constraints

Manually provide rules (or mine with AMIE+) and inject into loss function:

\[
\begin{align*}
\text{(a) Axioms extracted from YAGO3} & \\
\text{hasNeighor} & \equiv \text{hasNeighor}^- \\
\text{isMarriedTo} & \equiv \text{isMarriedTo}^- \\
\text{playsFor} & \equiv \text{isAffiliatedTo} \\
\text{isConnectedTo} & \equiv \text{isConnectedTo}^- \\
\text{(b) Axioms extracted from DBPEDIA} & \\
\text{assoc. band} & \equiv \text{assoc. musical artist} \\
\text{musical band} & \equiv \text{musical artist}
\end{align*}
\]

We extend \( \mathcal{L} \) with the regularization term \( \mathcal{R}_S \):

\[
\mathcal{L}_S(\Theta) = \mathcal{L}(\Theta) + \lambda \mathcal{R}_S(\Theta)
\]

\( \lambda = \infty \) hard constraints

\( \lambda = 0 \) original model

\[
\mathcal{R}_S(\Theta) \triangleq \sum_{p \equiv q \in \mathcal{A}_1} D [r_p \parallel r_q] + \sum_{p \equiv q^- \in \mathcal{A}_2} D [r_p \parallel \Phi(r_q)]
\]

\( D[\cdot\parallel\cdot] = ||\cdot - \cdot||_2^2 \): Divergence measure

\( \Phi(\cdot) \): Model-dependent transformation

\( \mathcal{A}_1 \): equivalent axioms set

\( \mathcal{A}_2 \): inverse axioms set

[Minervini et al. 2017]
KGE & Neuro-Symbolic Reasoning: Neural Theorem Provers (NTP)

• Rule-based models + KGE

• Interplay of KGE strengths (good generalization power, scalability) with rule-based interpretability (“small data” capabilities).

• NTP implement reasoning (e.g. backward chaining) in fully differentiable architectures
  • Symbols replaced by embeddings
  • Compare embeddings in Prolog backward chaining instead of matching symbols

[Rocktäschel et al. 2017]
[Minervini et al. 2020]
Interplay with Other Reasoning Regimes: Analogical Reasoning

ANALOGY  [Liu et al 2017]

• Models analogical structures in multi-relational embeddings
• “Differentiable” analogical reasoning combined with KGE models
Answering Complex Queries

**Query2box**: reasoning over Knowledge Graphs in a vector space using box embeddings to answer complex queries.

[Ren et al. 2020]
Outline

Theoretical Overview 1h 30m

- Introduction
- Anatomy of a Knowledge Graph Embedding Model
- Evaluation Protocol and Metrics
- Advanced KGE Topics
- Open Research Questions

Applications 15 m

Software Ecosystem 15 m

Hands-on Sessions 1h 15m
Open Research Questions

MORE EXPRESSIVE MODELS
Model KG regularities and dependencies while keeping runtime/space complexity low.

SUPPORT FOR MULTIMODALITY
Node and edge attributes, time-awareness still in their infancy.

ROBUSTNESS & INTERPRETABILITY
Techniques to dissect, investigate, explain, and protect from adversarial attacks.

BETTER BENCHMARKS
Agreed-upon fair evaluation protocols, novel datasets.

BEYOND LINK PREDICTION
Multi-path predictions, adoption in larger differentiable architectures to inject background knowledge from graphs.

NEURO-SYMBOLIC INTEGRATION
Integrate KGE non-differentiable reasoning regimes to get the best of different worlds.
Outline

Theoretical Overview
- Introduction
- Anatomy of a Knowledge Graph Embedding Model
- Evaluation Protocol and Metrics
- Advanced KGE Topics
- Open Research Questions

Q&A

Applications

Software Ecosystem

Hands-on Sessions
References


Outline

Theoretical Overview
- Introduction
- Anatomy of a Knowledge Graph Embedding Model
- Evaluation Protocol and Metrics
- Advanced KGE Topics
- Open Research Questions

Applications

Software Ecosystem

Hands-on Sessions

kge-tutorial-ecai2020.github.io
Industrial applications:

**Pharmaceutical Industry:**
Drug Side-effects Prediction

**Human Resources:**
Career Paths Prediction

**Products:**
Product Recommendation

**Food & Beverage:**
Flavor Combinations
Drug Development

• Drug Development is a time consuming and expensive process which ranges from gene identification, identifying a compound to target the gene, and finally experimentation on animals and humans.

• The initial step of identification of gene/drug takes several years and if not identified correctly may result in loss of time and money.

• “Drug Developers” identify the genes/drugs by reading the latest research before proceeding with experimentation. But it is highly dependent on the experience of the person.
Drug Development

Pharmaceutical Industry:
Drug Discovery
Drug Side Effect Prediction
Determining risk factors
Human Resource

• Technology is evolving at an extremely fast pace. People need to learn new skills to be relevant in the market.
• Due to automation, a lot of roles are becoming obsolete and companies are forced to lay off people.

KGEs can be used for following tasks:
• Suggest new technology/tasks for career progression.
• Recommend similar roles within the organization when existing role becomes obsolete.
Human Resource

Human Resources: Employee Career Progression or Transition
Product Recommendation

KGEs can leverage relation between customers and products. KGEs can be used for following tasks:

- Recommend new products to customers
- Group customers based on their purchase history
Product Recommendation

Retail:
Product Recommendation
Customer Grouping
Food & Beverage

Manufacturing/Retail:
- Product reformulation
- Adapting to consumer trends
Product Reformulation:
- Item substitution based on embedding proximity
Item Recommendation

• Use vector algebra to find latent region that satisfy input criteria

• Example:
  • “I want Indian recipes that contain garlic and tomato”
  • nearest(avg(avg(GARLIC,TOMATO) – containsIngredient, India – recipeOrigin))

• Note: the above is pseudo-code, actual solutions will depend on model, data, fine-tuning, etc

• Alternatively use Bayesian optimization ..
Graph Construction

Datasets

Pancakes, baking powder, eggs, all-purpose flour, raisins, milk, white sugar
Hummus, olive oil, chickpeas, lemon juice, tahini, garlic, cumin

Data processing

Recipe 1, contains Garlic
Recipe 1, contains Lemon juice

Triples

Garlic, (E)-2-Phenyl-2-butenal, Arginine, D-Aspartic acid L-Leucine, epsilon-Polylysine, (±)-erythro-Isolucine, Potassium

Garlic, hasCompound, Arginine
Garlic, hasCompound, L-Leucine

(+)-delta-Cadinene, hasOdor, herbal, woody, thyme, wood, medicine, dry 1,3-Dithiane, 10451, roasted, alliaceous

(+)-delta-Cadinene, hasPathway, gossypol biosynthesis
(+)-delta-Cadinene, hasPathway, lacinilene C biosynthesis

Knowledge Graph
Further reading ..


Outline

Theoretical Overview
• Introduction
• Anatomy of a Knowledge Graph Embedding Model
• Evaluation Protocol and Metrics
• Advanced KGE Topics
• Open Research Questions

Applications

Software Ecosystem

Hands-on Sessions
Outline

Theoretical Overview

Applications

Software Ecosystem

- Introduction
- What is Out There?
- Libraries Comparison
  - Features
  - Scalability
  - SOTA Reproduced
  - Software Development
- Which Library Should I Use?
- Questions

Hands-on Sessions
KGE’s Software Universe

Date reported is first of the following: pre-release/release/tag, in case where there are none of these the date reported is either submission date of a published paper accompanying or induced from repository activity. This is the case for scikit-kge, OpenKE, LibKGE.
Libraries comparison

... and what we measured

- Features
- Scalability
- SOTA Reproduced
- Software Development
Features
Models
Pre-trained models
Other Features
...
## Models

<table>
<thead>
<tr>
<th></th>
<th>TransE</th>
<th>DistMult</th>
<th>ComplEx</th>
<th>TransH</th>
<th>TransD</th>
<th>TransR</th>
<th>RESCAL</th>
<th>HoLE</th>
<th>SimplE</th>
<th>Analogy</th>
<th>ConvKB</th>
<th>ConvE</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenKE</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>AmpliGraph</td>
<td>✔️</td>
<td>✔️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✔️</td>
<td>✗️</td>
<td>✗️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>PyTorch</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td></td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>BigGraph</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗️</td>
<td>✗️</td>
<td>✔️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td></td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>GraphVite</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>DGL-KE</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗️</td>
<td>✗️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗️</td>
<td>✗️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>PyKEEN</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Pykg2vec</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Lib-KGE</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>scikit-kge</td>
<td>✔️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Model</td>
<td>RotatE</td>
<td>QuatE</td>
<td>KG2E</td>
<td>NTN</td>
<td>ProjE</td>
<td>RGCN</td>
<td>TuckER</td>
<td>TransM</td>
<td>CP</td>
<td>Other models</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>--------</td>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
<td>------</td>
<td>--------</td>
<td>--------</td>
<td>------</td>
<td>-------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenKE</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AmpliGraph</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PyTorch</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BigGraph</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GraphVite</td>
<td>✔</td>
<td>✔</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DGL-KE</td>
<td>✔</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PyKEEN</td>
<td>✔</td>
<td>x</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>x</td>
<td>Complex/DistMultLiteral, ERMLP, StructuredEmbedding, SME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pykg2vec</td>
<td>✔</td>
<td>x</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>SLM, SME, ComplexN3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lib-KGE</td>
<td>✔</td>
<td>✔</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✔</td>
<td>✔</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>scikit-kge</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>ERMLP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Pre-trained Models

<table>
<thead>
<tr>
<th></th>
<th>WikiData dump</th>
<th>Freebase</th>
<th>Benchmark datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenKE</td>
<td>✔️ link</td>
<td>✔️ (fragment ?)</td>
<td>✗</td>
</tr>
<tr>
<td>AmpliGraph</td>
<td>✗</td>
<td>✗</td>
<td>✔️ (upon request)</td>
</tr>
<tr>
<td>PyTorch BigGraph</td>
<td>✔️ (full)</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>GraphVite</td>
<td>✔️ (Wikidata5m)</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>DGL-KE</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>PyKEEN</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Pykg2vec</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Lib-KGE</td>
<td>✔️ (Wikidata5M)</td>
<td>✗</td>
<td>✔️ link</td>
</tr>
<tr>
<td>scikit-kge</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

OpenKE requires submitting your name, email and organization before download.

? The size of this embedding suggest it is a fragment.
Other Features

• OpenKE
  • C++ implementation.

• AmpliGraph
  • Benchmarking Aid and pre-processing.
  • Formats: rdf, csv, ntriples.
  • Knowledge discovery API.
  • Visualization.
  • Model selection API.
  • Slack.
  • Colab Tutorials.

• PBG
  • High-level operators.
  • Scalability (partitioning, experimental GPU).

• DGL-KE
  • Scalability (partitioning with METIS, faster than GraphVite and PBG).

• PyKEEN
  • Incorporating multi-modal information.
  • Extensibility (wide range of interchangeable components).
  • Hyperparameters support (Optuna).

• Pykg2vec
  • Metrics summary plots.
  • Automatic discovery for hyperparameters.
  • Interactive results inspector.

• Lib-KGE
  • Hyper param support (includes Bayesian Optimization).
  • Resuming training.
  • Configuration via yaml.

• GraphVite
  • Command line interface.
  • Visualization.
  • Configuration via yaml.
  • Auto-deduction of hyperparameters.
  • Scalability (GPU-CPU hybrid).
  • Node Embedding API.
  • Input data parser.

*scikit-kge is not listed here as it was discontinued*
Scalability
## Scalability

<table>
<thead>
<tr>
<th>Core Framework</th>
<th>GPU</th>
<th>Distributed Execution CPU</th>
<th>Biggest Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenKE</td>
<td>✔️</td>
<td>✗</td>
<td>10^8 edges, 4x10^7 nodes [3]</td>
</tr>
<tr>
<td>AmpliGraph</td>
<td>✗️</td>
<td>✗️ (Coming)</td>
<td>10^8 edges, 10^6 nodes [1]</td>
</tr>
<tr>
<td>PyTorch BigGraph</td>
<td>✔️</td>
<td>✔️</td>
<td>2.4x10^{12} edges, 1.21x10^8 nodes [4]</td>
</tr>
<tr>
<td>GraphVite</td>
<td>✔️</td>
<td>✔️ (GPU-CPU)</td>
<td>1.8x10^{12} edges, 6.6x10^6 nodes [9]</td>
</tr>
<tr>
<td>DGL-KE</td>
<td>✔️</td>
<td>✔️</td>
<td>3.38x10^8 edges, 8.6x10^6 nodes [8]</td>
</tr>
<tr>
<td>PyKEEN</td>
<td>✔️</td>
<td>✗</td>
<td>-</td>
</tr>
<tr>
<td>Pykg2vec</td>
<td>✔️</td>
<td>✗</td>
<td>-</td>
</tr>
<tr>
<td>Lib-KGE</td>
<td>✔️</td>
<td>✗</td>
<td>-</td>
</tr>
<tr>
<td>scikit-kge</td>
<td>✗️</td>
<td>✗</td>
<td>-</td>
</tr>
</tbody>
</table>
SOTA Reproduced
## SOTA Reproduced

<table>
<thead>
<tr>
<th></th>
<th>OpenKE</th>
<th>PBG</th>
<th>AmpliGraph</th>
<th>GraphVite</th>
<th>DGL-KE</th>
<th>PyKEEN</th>
<th>Pykg2vec</th>
<th>Lib-KGE</th>
<th>scikit-kge</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOTA reproduced</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Models reported</td>
<td>8/10</td>
<td>2/4(^1)</td>
<td>6/6</td>
<td>6/6</td>
<td>6/6</td>
<td>0/22(^2)</td>
<td>10/22</td>
<td>9/9</td>
<td>5/5(^2)</td>
</tr>
</tbody>
</table>

\(^1\) page 8, Lerer et. al. 2019.
\(^2\) page 6, Nickel et. al. 2015.
\(?\) Not found.
Software Development
Software Development Metrics

- Documentation ([docstr-coverage](https://pydocstyle.readthedocs.io/en/latest/)) - [PEP 257](https://www.python.org/dev/peps/pep-0257/)
  - Counts number of functions, classes, methods, and modules that doesn't have docstrings.

- Tests ([coverage](https://coverage.readthedocs.io/en/latest/))
  - measures how many lines out of the executable lines were executed.

- Good practices ([pylint](https://pylint.readthedocs.io/en/latest/)) [PEP 8](https://www.python.org/dev/peps/pep-0008/)

  **Formula:** $10.0 - ((float(5 * error + warning + refactor + convention) / statement) * 10)$

- Code Complexity ([radon](https://www.synopsys.com/products/continuous-integration/code-complexity-analysis.html)) – McCabe Complexity

<table>
<thead>
<tr>
<th>Class</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>1-10</td>
<td>10-20</td>
<td>20-40</td>
<td>40+</td>
</tr>
<tr>
<td>Code</td>
<td>Well written and structured</td>
<td>Complex Code</td>
<td>Very Complex Code</td>
<td>Extremely Complex Code</td>
</tr>
<tr>
<td>Testability</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
<td>Not Testable</td>
</tr>
<tr>
<td>Maintenance Cost and Effort</td>
<td>Less</td>
<td>Medium</td>
<td>High</td>
<td>Very High</td>
</tr>
</tbody>
</table>
Software Development Metrics

Documentation Coverage [%]

Tests Coverage [%]

Good Practices PEP8 (max 10)

Avg Code Complexity

- OpenKE
- AmpliGraph
- PBG
- Graph-Vite
- PyKEEN
- Pykg2vec
- Lib-KGE
- scikit-kge
Which Library Should I Use?

Align your choice with:

- Your task and time that you have for learning the library.
- Your experience.
- Framework the library supports Pytorch/Tensorflow/other?.
- Consider features like scalability, community support, user-friendliness, maturity of the project, accuracy, and supported addons.
- Finally: The choice is yours.

Use tools like github-statistics to support yourself.
Thank you!

Knowledge Graph Embeddings: From Theory to Practice

Software Ecosystem

adrianna.janik@accenture.com
Resources:

- AmpliGraph
- Libkge
- Graphvite
- DGL-KE
- Pykeen
- Pykg2vec
- OpenKE
- scikit-kge
- PyTorch-BigGraph
- github-statistics
- Article on how to compare repos
Outline

Theoretical Overview 1h 30m
• Introduction
• Anatomy of a Knowledge Graph Embedding Model
• Evaluation Protocol and Metrics
• Advanced KGE Topics
• Open Research Questions

Applications 15 m

Software Ecosystem 15 m

Hands-on Session 1h 15m

bit.ly/kge-tutorial
Knowledge Graph Embeddings: From Theory to Practice

Luca Costabello
Accenture Labs
@lukostaz

Sumit Pai
Accenture Labs
@sumitppai

Nicholas McCarthy
Accenture Labs
@nickpmcc

Adrianna Janik
Accenture Labs
@adri_janik

Slides & Colab: kge-tutorial-ecai2020.github.io
♫ by La Fascination