Knowledge Graph Embeddings meet Symbolic Schemas or: what do they Actually Learn?

Heiko Paulheim
University of Mannheim
Brief Introduction

Pre PhD Years
2006

PhD Years
2008 2011

PostDoc Years
2013 2014

Assistant Professor
2017

Full Professor

SDType
Kare
KoKI
 RDF2vec
ReNewRS

SoKNOS
Ontology-based Application Integration

FeGeLOD
Feature Generation from Linked Open Data

ADiWa
Ontologies-Based Business Integration

MELT

TECHNISCHE UNIVERSITÄT
DARMSTADT

Hochschule Darmstadt
University of Applied Sciences

UNIVERSITY OF MANNHEIM

SAP
The Beginning of my KG Embedding Journey

• Using Open (RDF) Data for improving data mining
Graphs vs. Vectors

• Data Science tools for prediction etc.
  – Python, Weka, R, RapidMiner, …
  – Algorithms that work on vectors, not graphs

• Bridges built over the past years:
    Python KG Extension (2021)
Graphs vs. Vectors

- Transformation strategies (aka *propositionalization*)
  - e.g., types: type_horror_movie=true
  - e.g., data values: year=2011
  - e.g., aggregates: nominations=7
Graphs vs. Vectors

• Observations with simple propositionalization strategies
  – Even simple features (e.g., add all numbers and types) can help on many problems
  – More sophisticated features often bring additional improvements
    • Combinations of relations and individuals
      – e.g., movies directed by Steven Spielberg
    • Combinations of relations and types
      – e.g., movies directed by Oscar-winning directors
    • ...
  – But
    • The search space is enormous!
    • Generate first, filter later does not scale well
Towards RDF2vec

• Excursion: word embeddings
  – word2vec proposed by Mikolov et al. (2013)
  – predict a word from its context or vice versa
    • Idea: similar words appear in similar contexts, like
      – Jobs, Wozniak, and Wayne founded Apple Computer Company in April 1976
      – Google was officially founded as a company in January 2006
  – usually trained on large text corpora
    • projection layer: embedding vectors
RDF2vec in a Nutshell

• Basic idea:
  – extract random walks from an RDF graph:
    Mulholland Dr. director David Lynch nationality US
  – feed walks into word2vec algorithm

• Order of magnitude (e.g., DBpedia)
  – ~6M entities (“words”)
  – start up to 500 random walks per entity, length up to 8
    → corpus of >20B tokens

• Result:
  – entity embeddings
  – most often outperform other propositionalization techniques

Ristoski and Paulheim (2016): RDF2vec: RDF graph embeddings for data mining
The End of Petar’s PhD Journey…

• …and the beginning of the RDF2vec adventure
Why does RDF2vec Work?

- Example: PCA plot of an excerpt of a cities classification problem
  - From cities classification task in the embedding evaluation framework by Pellegrino et al.
Why does RDF2vec Work?

- In downstream machine learning, we usually want class separation
  - to make the life of the classifier as easy as possible
- Class separation means
  - Similar entities (i.e., same class) are projected closely to each other
  - Dissimilar entities (i.e., different classes) are projected far away from each other
Why does RDF2vec Work?

- Observation: close projection of similar entities
  - Usage example: content-based recommender system based on k-NN

Ristoski and Paulheim (2016): RDF2vec: RDF graph embeddings for data mining
Embeddings for Link Prediction

- RDF2vec observations
  - similar instances form clusters, direction of relation is ~stable
  - link prediction by analogy reasoning (Japan – Tokyo ≈ China – Beijing)

Ristoski & Paulheim: RDF2vec: RDF Graph Embeddings for Data Mining. ISWC, 2016
Embeddings for Link Prediction

• In RDF2vec, relation preservation is a by-product
• TransE (and its descendants): direct modeling
  – Formulates RDF embedding as an optimization problem
  – Find mapping of entities and relations to $\mathbb{R}^n$ so that
    • across all triples $<s,p,o>$
      $\sum ||s+p-o||$ is minimized
    • try to obtain a smaller error for existing triples
      than for non-existing ones

Fan et al.: Learning Embedding Representations for Knowledge Inference on Imperfect and Incomplete Repositories. WI 2016
Link Prediction vs. Node Embedding

- Hypothesis:
  - Embeddings for link prediction also cluster similar entities
  - Node embeddings can also be used for link prediction

Portisch, Heist, Paulheim: Knowledge Graph Embedding for Data Mining vs. Knowledge Graph Embedding for Link Prediction - Two Sides of the Same Coin? Semantic Web Journal, 2022
Embeddings are Here to Stay

![Graph showing the increase in publications related to embeddings from 2014 to 2023.](image)

Source: https://app.dimensions.ai
Exported: September 11, 2023
Criteria: 'knowledge graph embedding' in title and abstract.

© 2023 Digital Science and Research Solutions Inc. All rights reserved. Non-commercial redistribution / external re-use of this work is permitted subject to appropriate acknowledgement. This work is sourced from Dimensions® at www.dimensions.ai.
The 2009 Semantic Web Layer Cake

- User Interface and Applications
- Trust
- Proof
- Unifying Logic
  - Query: SPARQL
  - Ontology: OWL
  - Rules: RIF
  - Schema: RDF-S
- Data Interchange: RDF
- Data Interchange: XML
- URI
- Unicode
The 2018 Semantic Web Layer Cake

Embeddings

User Interface and Applications

Data Interchange: RDF

Data Interchange: XML

URI

Unicode
The 2023 Semantic Web Layer Cake

User Interface and Applications

LLMs
The People’s Front of Embeddings Keeps Asking:

What have the ontologists ever done for us?

What have the ontologists ever done for us?
Do we no Longer Need Semantic Schemas?

- Schemas/ontologies are sometimes used for
  - injection in embedding creation process, e.g.
    - **Graph preprocessing (inference)**
    - Creation of non-trivial negatives
    - Ontological compliance as loss term
    - **Pre-training on protograph**
  - analysis of results
    - Measuring ontological compliance (e.g., Sem@k)
    - **Quantifying representational capability**
  - ...

...
To Materialize or Not to Materialize?

May I ask you a question?

Sure, go ahead!
Rumor has it that RDF2vec performs worse if you run a reasoner to add inferences to the graph first...
To Materialize or Not to Materialize?

I know it sounds counter intuitive...

Hmmm...
To Materialize or Not to Materialize?

Hmmm… sounds reasonable. (Pun intended)

Okay, there might be an explanation…
To Materialize or Not to Materialize?

We need more beer experiments
Back Home...

We need more beer experiments

OK, let’s go!
Experimental Setup

Iana and Paulheim (2020): More is not always better: The negative impact of a-box materialization on RDF2vec knowledge graph embeddings
Experimental Results

- Classification: unmaterialized is better in 60/80 cases
- Regression: unmaterialized is better in 39/60 cases
- Entity similarity: unmaterialized is better in 16/20 cases
- Entity relatedness: unmaterialized is better in 13/20 cases

- But: document similarity: materialized is always better
  - task has a very different nature
  - more heterogeneity

Iana and Paulheim (2020): More is not always better: The negative impact of a-box materialization on RDF2vec knowledge graph embeddings
To Materialize or not to Materialize?

- Explanation 1: materialization skews property distributions

Iana and Paulheim (2020): More is not always better: The negative impact of a-box materialization on RDF2vec knowledge graph embeddings
To Materialize or not to Materialize?

- Explanation 2 is a bit more complex...
- Thought experiment:
  - DBpedia mostly does not include persons’ gender
  - learn classifier for gender
- Spouse is a symmetric property, but...
  - distribution is highly uneven
  - 80% of all subjects of *spouse* are women

Ayda_Field spouse Robbie_Williams.


<table>
<thead>
<tr>
<th>Attribute</th>
<th>% Men</th>
<th>% Women</th>
<th>$\chi^2$</th>
<th>$w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>birthName</td>
<td>4.01</td>
<td>11.46</td>
<td>4.84**</td>
<td>0.81</td>
</tr>
<tr>
<td>careerStation</td>
<td>8.95</td>
<td>1.13</td>
<td>6.84**</td>
<td>0.94</td>
</tr>
<tr>
<td>deathDate</td>
<td>32.82</td>
<td>19.35</td>
<td>5.53</td>
<td>0.64</td>
</tr>
<tr>
<td>deathYear</td>
<td>44.68</td>
<td>25.45</td>
<td>8.28**</td>
<td>0.66</td>
</tr>
<tr>
<td>formerTeam</td>
<td>4.40</td>
<td>0.24</td>
<td>3.94**</td>
<td>0.97</td>
</tr>
<tr>
<td>numberOfMatches</td>
<td>8.60</td>
<td>1.06</td>
<td>6.61**</td>
<td>0.94</td>
</tr>
<tr>
<td>occupation</td>
<td>12.52</td>
<td>23.28</td>
<td>4.97</td>
<td>0.68</td>
</tr>
<tr>
<td>position</td>
<td>13.62</td>
<td>1.68</td>
<td>10.46**</td>
<td>0.94</td>
</tr>
<tr>
<td>spouse</td>
<td>1.56</td>
<td>6.86</td>
<td>4.10**</td>
<td>0.88</td>
</tr>
<tr>
<td>team</td>
<td>14.06</td>
<td>1.97</td>
<td>10.39**</td>
<td>0.93</td>
</tr>
<tr>
<td>title</td>
<td>9.17</td>
<td>19.65</td>
<td>5.59</td>
<td>0.73</td>
</tr>
<tr>
<td>years</td>
<td>8.95</td>
<td>1.12</td>
<td>6.84**</td>
<td>0.94</td>
</tr>
</tbody>
</table>
To Materialize or not to Materialize?

• Thought experiment: learn classifier for gender
• Spouse is a symmetric property, but…
  – 80% of all subjects of spouse are women
• Assume that an embedding captures that information
  → a downstream classifier can reach >80% accuracy
• On the other hand
  – Materialization completely erases that information

• Bottom line: missing information can be a signal
  – Machine learning terminology: MAR vs. MNAR

Iana and Paulheim (2021): More is not Always Better: The Negative Impact of A-box Materialization on RDF2vec Knowledge Graph Embeddings
Protographs

• Schema of a graph
  – Can be seen as a “prototype” for the actual instances
  – Allows for instantiation of a smaller KG prototype
  • Very fast training

Nicolas Hubert, Heiko Paulheim, Pierre Monnin, Armelle Brun and Davy Monticolo: Schema First! Learn Versatile Knowledge Graph Embeddings by Capturing Semantics with MASCHInE. Under Review.
Protographs

- After initialization:
  - Transfer prototype embeddings to actual instances
  - Refine the embedding

Nicolas Hubert, Heiko Paulheim, Pierre Monnin, Armelle Brun and Davy Monticolo: Schema First! Learn Versatile Knowledge Graph Embeddings by Capturing Semantics with MASCHInE. Under Review.
Protographs

- Results
  - Class separation gets better
  - Gains on node clustering & classification, slight degradation on LP

Nicolas Hubert, Heiko Paulheim, Pierre Monnin, Armelle Brun and Davy Monticolo: Schema First! Learn Versatile Knowledge Graph Embeddings by Capturing Semantics with MASCHInE. Under Review.
Class Separation, Clustering, etc.

- What do Knowledge Graph Embeddings actually learn?

Nicolas Hubert, Heiko Paulheim, Pierre Monnin, Armelle Brun and Davy Monticolo: Schema First! Learn Versatile Knowledge Graph Embeddings by Capturing Semantics with MASCHInE. Under Review.
Close Projection of Similar Entities

• What does similar mean?
### Similarity vs. Relatedness

- Closest 10 entities to *Angela Merkel* in different vector spaces

<table>
<thead>
<tr>
<th>RDF2vec</th>
<th>TransE-L1</th>
<th>TransE-L2</th>
<th>TransR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joachim Gauck</td>
<td>Gerhard Schröder</td>
<td>Gerhard Schröder</td>
<td>Sigmar Gabriel</td>
</tr>
<tr>
<td>Norbert Lammert</td>
<td>James Buchanan</td>
<td>Helmut Kohl</td>
<td>Frank-Walter Steinmeier</td>
</tr>
<tr>
<td>Stanislaw Tillich</td>
<td>Neil Kinnock</td>
<td>Konrad Adenauer</td>
<td>Philipp Rösler</td>
</tr>
<tr>
<td>Andreas Voßkuhle</td>
<td>Nicolas Sarkozy</td>
<td>Helmut Schmidt</td>
<td>Gerhard Schröder</td>
</tr>
<tr>
<td>Berlin</td>
<td>Joachim Gauck</td>
<td>Werner Faymann</td>
<td>Joachim Gauck</td>
</tr>
<tr>
<td>German language</td>
<td>Jacques Chirac</td>
<td>Alfred Gusenbauer</td>
<td>Christian Wulff</td>
</tr>
<tr>
<td>Germany</td>
<td>Jürgen Trittin</td>
<td>Kurt Georg Kiesinger</td>
<td>Guido Westerwelle</td>
</tr>
<tr>
<td>federalState</td>
<td>Sigmar Gabriel</td>
<td>Philipp Scheidemann</td>
<td>Helmut Kohl</td>
</tr>
<tr>
<td>Social Democratic Party</td>
<td>Guido Westerwelle</td>
<td>Ludwig Erhard</td>
<td>Jürgen Trittin</td>
</tr>
<tr>
<td>deputy</td>
<td>Christian Wulff</td>
<td>Wilhelm Marx</td>
<td>Jens Böhmken</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RotatE</th>
<th>DistMult</th>
<th>RESCAL</th>
<th>ComplEx</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pontine raphe nucleus</td>
<td>Gerhard Schröder</td>
<td>Gerhard Schröder</td>
<td>Gerhard Schröder</td>
</tr>
<tr>
<td>Jonathan W. Bailey</td>
<td>Milan Truban</td>
<td>Kurt Georg Kiesinger</td>
<td>Diana Mészáros</td>
</tr>
<tr>
<td>Zokwang Trading</td>
<td>Maud Cuney Hare</td>
<td>Helmut Kohl</td>
<td>Francis M. Bator</td>
</tr>
<tr>
<td>Steven Hill</td>
<td>Tristan Matthiae</td>
<td>Annemarie Huber-Hotz</td>
<td>William B. Bridges</td>
</tr>
<tr>
<td>Chad Kreuter</td>
<td>Gerda Hasselfeldt</td>
<td>Wang Zhaoguo</td>
<td>Mette Vestergaard</td>
</tr>
<tr>
<td>Fred Hibbard</td>
<td>Faustino Sainz Muñoz</td>
<td>Franz Vranitzky</td>
<td>Ivan Rosenqvist</td>
</tr>
<tr>
<td>Mallory Ervin</td>
<td>Joachim Gauck</td>
<td>Bögdon Klich</td>
<td>Edward Clouston</td>
</tr>
<tr>
<td>Paulinho Kobayashi</td>
<td>Carsten Linnemann</td>
<td>Irsen Küçük</td>
<td>Antonio Capuzzi</td>
</tr>
<tr>
<td>Fullmetal Alchemist and the Broken Angel</td>
<td>Norbert Blüm</td>
<td>Helmut Schmidt</td>
<td>Steven J. McAuliffe</td>
</tr>
<tr>
<td>Archbishop Dorotheus of Athens</td>
<td>Neil Hood</td>
<td>Mao Zedong</td>
<td>Jenkin Coles</td>
</tr>
</tbody>
</table>

Portisch et al. (2022): Knowledge Graph Embedding for Data Mining vs. Knowledge Graph Embedding for Link Prediction - Two Sides of the Same Coin?
Back to Class Separation

- What is a class?
  - e.g., cities per se
  - e.g., cities in Norway
  - e.g., cities in Norway above 500k inhabitants
- Or something different, such as
  - e.g., everything located in Oslo
  - e.g., everything Oslo is known for
Back to Class Separation

- Observation: there are different kinds of classes:
  - Classes of objects of the same category (e.g., cities)
    → those are *similar*
  - Classes of objects of different categories (e.g., buildings, dishes, organizations, persons)
    → those are *related*
Intermediate Observation

- In most vector spaces of link prediction embeddings (TransE etc.): proximity ~ similarity
- In RDF2vec embedding space: proximity ~ a mix of similarity and relatedness
So... why does RDF2vec Work Then?

- Recap: downstream ML algorithms need class separation
  - but RDF2vec groups items by similarity and relatedness
- Why is RDF2vec still so good at classification?
Example

- It depends on the classification problem at hand!
  - Cities vs. countries
  - Places in Europe vs. places in Asia
So… why does RDF2vec Work Then?

- Many downstream classification tasks are **homogeneous**
  - e.g., classifying cities in different subclasses
- For homogeneous entities:
  - relatedness provides finer-grained distinctions
Variants of RDF2vec

- Observation: order matters!
- Recap word embeddings:
  - Jobs, Wozniak, and Wayne founded Apple Computer Company in April 1976
  - Google was officially founded as a company in January 2006
- Graph walks:
  - Hamburg → country → **Germany** → leader → Angela_Merkel
  - Germany → leader → **Angela_Merkel** → birthPlace → Hamburg
  - Hamburg → leader → **Peter_Tschentscher** → residence → Hamburg

Portisch and Paulheim: Putting RDF2vec in Order. ISWC, 2021
Variants of RDF2vec

• Which parts of a walk denote what?
  – Hamburg → country → Germany → leader → Angela_Merkel
  – Germany → leader → Angela_Merkel → birthPlace → Hamburg
  – Hamburg → leader → Peter_Tschentscher → residence → Hamburg
  – California → leader → Gavin_Newsom → birthPlace → San_Francisco

• Common predicates (leader, birthPlace)
  – Similar entities

• Common entities (Hamburg)
  – Related entities
  – For same-class entities: similar entities!

• Approach: entity walks (e-walks) and property walks (p-walks)

The RDF2vec Zoo

• We now have an entire zoo of RDF2vec variants
  – SG vs. CBOW
  – Order-aware vs. unordered ("classic")
  – Walk variants
Which Classes can be Learned with RDF2vec?

• We already saw that there are different notions of classes
• Idea: compile a list of class definitions as a benchmark
  – Classes are expressed as DL formulae, e.g.
  – ∃r.T, e.g. Class \textit{person with children}
  – ∃r.{e}, e.g.: Class \textit{person born in New York City}
  – ∃R.{e}, e.g., Class \textit{person with any relation to New York City}
  – ∃r.C, e.g., Class \textit{person playing in a basketball team}
  – …
Which Classes can be Learned with RDF2vec?

• Formulating hypotheses
  – e.g., $\exists r . T$, cannot be learned when using e-walks

• Testing hypotheses
  – using queries against DBpedia

• ...

The DLCC DBpedia Gold Standard

• Six classes (person, book, city, movie, album, species)
• Twelve test cases
  – Sometimes also with “harder” negatives
• Three sizes per test case (50, 500, 5,000 examples)
  – Each is a balanced binary classification problem

• >200 hand written SPARQL queries
• Dataset and code available online
Hypotheses (Overview)

• Different patterns require different signals, e.g.,
  – Specific relations (visible to classic and p-walks)
  – Specific entities (visible to classic and e-walks)
  – Distinguishing subject and object (only possible for oa variants)
  – ...and mixes of those
Which Classes can be Learned with RDF2vec?

- Formulating hypotheses
  - e.g., $\exists r . T$, cannot be learned when using e-walks
- Testing hypotheses
  - using queries against DBpedia
- Seeing surprises
  - e.g., models trained on e-walks can reach ~90% accuracy in that case
Experiments on DBpedia Gold Standard

- Most hypotheses could not be confirmed
- All problems are learnable with an accuracy >75%
  - i.e., significantly better than guessing
- Also LP embeddings such as TransE work surprisingly well
Experiments on DBpedia Gold Standard

- Challenge: isolating effects
  - Let’s consider, $\exists r. T$: e.g. $\exists$almaMater.T
  - In theory, we should not be able to learn this with e-walks
  - Frequent entities in the neighborhoods of positive examples:
    - Politician (3k examples)
    - Bachelor of Arts (3k examples)
    - Harvard Law School (2k examples)
    - Lawyer (2k examples)
    - Northwestern University (2k examples)
    - Harvard University (2k examples)
    - Doctor of Philosophy (2k examples)
    - ...
  - Those signals are visible to e-walks!
Which Classes can be Learned with RDF2vec?

• Maybe, DBpedia is not such a great testbed
  – Hidden patterns, e.g., for relation cooccurrence
  – Many inter-pattern dependencies
  – Information not missing at random

• Possible solution:
  – Synthetic knowledge graphs!
  – First experiments show better visibility of expected effects
The DLCC Synthetic Gold Standard

- Same twelve test cases as before
- Synthesize a knowledge graph for each test case
  - Create an ontology
  - Create positive examples
  - Create negative examples (double check for accidental positives)
- Test bed
  - 12 different classification problems, 1k positives/negatives each
  - Ontology and graph structure are similar to DBpedia
Experiments on DLCC Synthetic Gold Standard

• Same twelve test cases as before
• Synthesize a knowledge graph for each test case
  – Create an ontology
  – Create positive examples
  – Create negative examples (double check for accidental positives)
• Test bed
  – 12 different classification problems, 1k positives/negatives each
  – Ontology and graph structure are similar to DBpedia
Experiments on DLCC Synthetic Gold Standard

- Hypotheses can be mostly confirmed
  - Quantified restrictions (tc09-tc12) are very badly learned by all approaches (as expected)
  - tc06 is extremely well learned by LP embeddings
    - Classifying r.{e} is, in fact, classic link prediction/triple scoring
    - RDF2vec_{oa} variant is not superior on synthetic data

Portisch&Paulheim: The DLCC Node Classification Benchmark for Analyzing Knowledge Graph Embeddings. ISWC 2022.
Alternatives to Understand KG Embeddings

- Approach 1: learn symbolic interpretation function for dimensions
- Each dimension of the embedding model is a target for a separate learning problem
- Learn a function to explain the dimension
- E.g.: \( y \approx -|\exists \text{character} . \text{Superhero}| \)
- Just an approximation used for explanations and justifications
Alternatives to Understand KG Embeddings

- **Approach 2:** learn symbolic substitute function for similarity function

<table>
<thead>
<tr>
<th>RDF2vec</th>
<th>TransE-L1</th>
<th>TransE-L2</th>
<th>TransR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joachim Gauck</td>
<td>Gerhard Schröder</td>
<td>Gerhard Schröder</td>
<td>Sigmar Gabriel</td>
</tr>
<tr>
<td>Norbert Lammert</td>
<td>James Bachmann</td>
<td>Helmut Kohl</td>
<td>Frank-Walter Steinmeier</td>
</tr>
<tr>
<td>Stanislaw Tillich</td>
<td>Neil Kinross</td>
<td>Konrad Adenauer</td>
<td>Philipp Röver</td>
</tr>
<tr>
<td>Andreas Vollkholte</td>
<td>Nicolas Sarkozy</td>
<td>Helmut Schmidt</td>
<td>Gerhard Schröder</td>
</tr>
<tr>
<td>Berlin</td>
<td>Joachim Gauck</td>
<td>Werner Faymann</td>
<td>Joachim Gauck</td>
</tr>
<tr>
<td>German language</td>
<td>Jacques Chirac</td>
<td>Alfred Geesebauer</td>
<td>Christian Wulff</td>
</tr>
<tr>
<td>Germany</td>
<td>Jürgen Tintin</td>
<td>Kurt Georg Kiesinger</td>
<td>Guido Westerwelle</td>
</tr>
<tr>
<td>federalState</td>
<td>Sigmar Gabriel</td>
<td>Philipp Scheidenmann</td>
<td>Helmut Kohl</td>
</tr>
<tr>
<td>Social Democratic Party</td>
<td>Guido Westerwelle</td>
<td>Ludwig Enhardt</td>
<td>Jürgen Tintin</td>
</tr>
<tr>
<td>deputy</td>
<td>Christian Wulff</td>
<td>Wilhelm Marx</td>
<td>Jens Bohmns</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RotatE</th>
<th>DiuMult</th>
<th>RESCAL</th>
<th>ComplEx</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pontine raphe nucleus</td>
<td>Gerhard Schröder</td>
<td>Gerhard Schröder</td>
<td>Gerhard Schröder</td>
</tr>
<tr>
<td>Jonathan W. Bailey</td>
<td>Milan Tsuban</td>
<td>Kurt Georg Kiesinger</td>
<td>Diana Mészáros</td>
</tr>
<tr>
<td>Zhkwang Trading</td>
<td>Maud Coney Blaze</td>
<td>Helmut Kohl</td>
<td>Francis M. Bates</td>
</tr>
<tr>
<td>Steven Hill</td>
<td>Tristan Matthae</td>
<td>Anemarke Huber-Hotz</td>
<td>William B. Bridges</td>
</tr>
<tr>
<td>Chad Kreuter</td>
<td>Gerda Hausfeld</td>
<td>Wang Zhaoqiao</td>
<td>Mette Vestygaard</td>
</tr>
<tr>
<td>Fred Hibbard</td>
<td>Faustino Sainz Matias</td>
<td>Franz Vranitzky</td>
<td>Ivan Rosepist</td>
</tr>
<tr>
<td>Malheor Ervin</td>
<td>Joachim Gauck</td>
<td>Bogdan Klich</td>
<td>Edward Clouston</td>
</tr>
<tr>
<td>Paulino Kobayashi</td>
<td>Carsten Linnemann</td>
<td>Iren Köck</td>
<td>Antonio Capuzzi</td>
</tr>
<tr>
<td>Fullmetal Alchemist and the Broken Angel</td>
<td>Norbert Blüm</td>
<td>Helmut Schmidt</td>
<td>Steven J. McAuliffe</td>
</tr>
<tr>
<td>Archbishop Dorotheus of Athens</td>
<td>Neil Hood</td>
<td>Mao Zeqiong</td>
<td>Jonkin Coles</td>
</tr>
</tbody>
</table>

Right hand side picture: RelFinder (https://interactivesystems.info/developments/relnfnder)
Alternatives to Understand KG Embeddings

• Approach 3: generate symbolic interpretations for individual predictions
  – Inspired by LIME:
    • Generate perturbed examples
    • Label them using embedding+downstream classifier
    • Learn symbolic model on this labeled set
  – Good news:
    • RDF2vec can, in principle, create embeddings for unseen entities
    • Those can be used to classify perturbed examples
Summary

- Knowledge Graph Embeddings & Semantic Schemas
- We have seen today
  - How materialization and inference do (not) help
  - How schemas can be used for efficient initialization of KGE models
  - How to explore what embedding models can actually learn
More on RDF2vec

- Collection of
  - Implementations
  - Pre-trained models
  - >75 use cases in various domains
Thank you!

🌐 http://www.heikopaulheim.com

🐦 @heikopaulheim
Knowledge Graph Embeddings meet Symbolic Schemas or: what do they Actually Learn?

Heiko Paulheim
University of Mannheim