ML-Based Knowledge Graph Curation: Current Solutions and Challenges

Laure Berti-Equille

IRD Montpellier
Aix-Marseille University, CNRS, LIS, DIAMS
France
laure.berti@ird.fr
http://pageperso.lif.univ-mrs.fr/~laure.berti/
Data Quality Problems in KBs

What can go wrong?

In DL:
- Invalid ABox: Class (concept), Property (role), Constant (individual)
- Invalid TBox: Set of axioms (Bad ontology design defining relationships: hierarchies, domains, ranges, etc.)

In RDF:
Invalid Triple:
  <subject, property, object>

In KG:
Invalid Fact:
  < head, relation, tail >
Invalid Reference to Extra-Information
  - Mismatch of entity description
  - Ambiguities in context mention

DATA QUALITY PROBLEMS

<table>
<thead>
<tr>
<th>TYPE</th>
<th>CARDINALITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing data</td>
<td>Single-Point</td>
</tr>
<tr>
<td>Anomalous data</td>
<td>Collection</td>
</tr>
<tr>
<td>Duplicate data</td>
<td></td>
</tr>
<tr>
<td>Inconsistent data</td>
<td></td>
</tr>
<tr>
<td>Obsolete data</td>
<td></td>
</tr>
<tr>
<td>Incorrect data</td>
<td></td>
</tr>
</tbody>
</table>

DETECTION/CORRECTION MODE

Manual Inspection:
- Expert and Human-In-the-Loop
- Find-Fix-Verify Crowdsourcing

Semi- or unsupervised techniques:
- Constraints, Rules, and Patterns
- Descriptive Statistics
- Model Inference and Machine Learning
Sources of errors in KB Construction/Population

- **Data Extraction**
  - Errors in unsupervised knowledge extraction from unstructured texts in open domain
  - Multi-lingual and cultural difficulties in information extraction
  - Identity problem due to context/description mismatch
  - Obsolescence

- **Entity Linking**
  - Accuracy of automatic data linking approaches and large-scale entity disambiguation

- **Knowledge Inference**
  - Inadequate knowledge representations (information loss)
  - Inadequacy of KG semantic embedding techniques for 1-N, N-1, and N-N relations

- **Knowledge Publishing**
  - Lack of automated large-scale knowledge verification and curation
  - Lack of KG completion explainability (provenance), comprehensiveness, and interpretability
Quality = Fitness for Use

Accuracy, Consistency, Freshness, Completeness, Uniqueness

Precision, Timeliness, Conciseness, Interpretability, Accessibility, Objectivity, Security, Relevance, Source Reputation, Understandability, Believability, Ease of use [...]

Up to 179 dimensions for Data Quality\(^1\) only 18 applicable to LOD\(^2\) with a dedicated ontology\(^3\)


\(^3\) Debattista, Lange, Auer - daQ, an Ontology for Dataset Quality Information LDOW2014
Research Context

1. Designing ML-based solutions for Data and Knowledge engineering is a very hot topic in DB community
2. Tsunami of Deep NN architectures and applications
Outline

Introduction
• Motivations
• Context
• Examples illustrating some relevant work

ML-based KG Curation
• KG refinement and ontology learning
• KG embedding
• KG completion
• Consistency checking and KG repairing

Concluding Remarks & Perspectives
Are all resources and KBs equally complete, accurate, up-to-date, and trustworthy?

Of course not!
Example 1. Completeness

<table>
<thead>
<tr>
<th>Class name</th>
<th>#Objects</th>
<th>#Properties</th>
<th>Class completeness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cats</td>
<td>133</td>
<td>2</td>
<td>0.00%</td>
</tr>
<tr>
<td>Fictional donkeys and some fact about them</td>
<td>14</td>
<td>3</td>
<td>2.38%</td>
</tr>
<tr>
<td>US Presidents</td>
<td>79</td>
<td>5</td>
<td>3.54%</td>
</tr>
<tr>
<td>States of Austria</td>
<td>9</td>
<td>2</td>
<td>16.67%</td>
</tr>
<tr>
<td>Cantons of Switzerland</td>
<td>26</td>
<td>3</td>
<td>6.41%</td>
</tr>
</tbody>
</table>

16 out of 46 known non-functional properties are complete.
Example 1 (Cont’ed).

KB Representativeness and Bias

Suppose you have the accurate and complete knowledge of the world-wide populations per city grouped into 4 categories: e.g. (<100k, [100k,500k], [500k,1M], >1M) and 4 KBs.

K_1 is more complete than K_2 but both are somehow biased toward one category K_1 and K_2 are not as representative as K_3 or K_4

- Pitoura, Tsoparas, Flouris, Fundulaki, Papadakis, Abiteboul, Weikum. On Measuring Bias in Online Information. SIGMOD Record, Vol.46 No.4, December 2017
### Example 2. KB Correctness

**Relational data quality problems**

*Nobel Laureates in Chemistry*

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution</th>
<th>Institution_City</th>
<th>DoB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skłodowska-Curie Marie</td>
<td>Institut Pasteur</td>
<td>Varsovie</td>
<td>07-11-1867</td>
</tr>
<tr>
<td>M. Curie</td>
<td>Pasteur Institute</td>
<td>Paris</td>
<td>1867-11-07</td>
</tr>
<tr>
<td>Melvin Calvin</td>
<td>UC Berkeley</td>
<td>Berkeley</td>
<td>1911-04-08</td>
</tr>
<tr>
<td>Marie Curien</td>
<td>Paris</td>
<td>Pasteur Institute</td>
<td>2007-11-07</td>
</tr>
<tr>
<td>Avram Hershko</td>
<td>NULL</td>
<td>Haifa</td>
<td>NULL</td>
</tr>
<tr>
<td>Ronald Hoffman</td>
<td>NULL</td>
<td>US</td>
<td>00000000</td>
</tr>
</tbody>
</table>
Example 2 (Cont’ed). KB Correctness

Knowledge Graph data problems
Nobel Laureates in Chemistry: Excerpt

Complex combination of:
- Missing links and entities
- Spurious links: existence, type, direction
- Erroneous entity name
- Errors in literal values with various degrees of severity:
  formatting, up-to-dateness, veracity issues
Example 3. Numerical Outliers

Rejection area: Data space excluding the area defined between 2% and 98% quantiles for X and Y

Rejection area based on:
Mahalanobis\_dist(cov(X,Y)) > \chi^2(.98,2)

(Classical Setting)

Bivariate Analysis

Multivariate Analysis

Legitimate outliers or data quality problems?
Example 3 (Cont’ed). Numerical Outliers in KG

Need for more approaches leveraging ontology, constraints or dependencies to improve outlier detection

Fig. 1: Example for subpopulation lattice for property population. Numbers to the upper right of a node give the number of instances fulfilling the constraint set. Dashed nodes would be pruned, the left one for too low KL divergence, the right one for not reducing the instance set further.

<table>
<thead>
<tr>
<th>Approach</th>
<th>elevation</th>
<th>height</th>
<th>populationTotal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outlier Detection</td>
<td>0.872</td>
<td>0.888</td>
<td>0.876</td>
</tr>
<tr>
<td>Cross-Checked Outlier Detection</td>
<td>0.861</td>
<td>0.891</td>
<td>0.941</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.745</td>
<td>0.847</td>
<td>0.847</td>
</tr>
<tr>
<td>Multi-lingual Baseline</td>
<td>0.669</td>
<td>0.509</td>
<td>0.860</td>
</tr>
</tbody>
</table>

Example 4. Veracity and Trustworthiness

ML-based approach for knowledge-based trust:
• Multi-Layer Model based on EM and Bayesian inference
• Distinguish extractor errors from source errors

Example 5: Up-to-dateness

Asynchronous Real World and KG evolution

Table 1. DBpedia - Classes and Properties

<table>
<thead>
<tr>
<th>Version</th>
<th>OWL Class</th>
<th>RDF Property</th>
<th>Object Prop.</th>
<th>Datatype Prop.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#</td>
<td>∆</td>
<td>(-)</td>
<td>(+)</td>
</tr>
<tr>
<td>3.2/3</td>
<td>174</td>
<td>720</td>
<td>384</td>
<td>336</td>
</tr>
<tr>
<td>3.4</td>
<td>204</td>
<td>30</td>
<td>-2</td>
<td>32</td>
</tr>
<tr>
<td>3.5</td>
<td>255</td>
<td>51</td>
<td>-6</td>
<td>57</td>
</tr>
<tr>
<td>3.6</td>
<td>272</td>
<td>17</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>3.7</td>
<td>319</td>
<td>47</td>
<td>-1</td>
<td>48</td>
</tr>
<tr>
<td>3.8</td>
<td>359</td>
<td>40</td>
<td>-1</td>
<td>41</td>
</tr>
<tr>
<td>3.9</td>
<td>529</td>
<td>170</td>
<td>-1</td>
<td>171</td>
</tr>
<tr>
<td>2014</td>
<td>683</td>
<td>154</td>
<td>-5</td>
<td>159</td>
</tr>
<tr>
<td>2015-04</td>
<td>735</td>
<td>52</td>
<td>-5</td>
<td>57</td>
</tr>
<tr>
<td>2015-10</td>
<td>739</td>
<td>4</td>
<td>-5</td>
<td>9</td>
</tr>
<tr>
<td>2016-04</td>
<td>754</td>
<td>15</td>
<td>0</td>
<td>15</td>
</tr>
</tbody>
</table>

Today’s DBpedia Ontology: 685 classes described by 2,795 properties


Outline

Introduction

• Motivations
• Context
• Examples illustrating some relevant work

ML-based KG Data Curation
ML-based Solutions for KG Curation

Knowledge Graph Refinement
Ontology Learning to learn a concept level description of a domain (e.g., Cities are Places)

Knowledge Extraction
Fact Extraction and Verification: Knowledge Fusion Methods

Completion of Knowledge Graphs
- Learning Embeddings
- Methods for Entity Linking & Link Prediction: classification, rank, probabilistic graph models, deep (reinforcement) learning

Error Detection and Repair in Knowledge Graphs
- Rule learning for detecting/correcting erroneous type assertions, relations or literal values
- User-guided repair with updates
GLUE: Learning to find similar ontological concepts

- Glue applies ML technique to find, for each concept node in a taxonomy, the most similar concept in the other taxonomy
- It applies the multi-learning approach of LSD (*Learning Source Description*)

---


Fig. 2. The GLUE Architecture
GLUE: Learning to find similar ontological concepts (2)

• It leverages the joint probability distribution:
  – \( P(A,B), P(A, \text{not}(B)), P(\text{not}(A),B), P(\text{not}(A),\text{not}(B)) \)

• ML is used to infer whether \( P(A,B) \) can be approximated with \( P(A \text{ intersect } B) \)
  – By defining a classifier for instances containing concept A (resp. B) and using it to classify instances of B (resp. A)

Fig. 3. Estimating the joint distribution of concepts A and B

Learning distributed representations of entities and relations of KG

• Linear models
  – Tensor product-based: RESCAL, DistMult, ComplEx, SimplE, TuckER

• Deep Learning or convolution
  – HypER, ConvE, ConKB, SLM, LFM, ER-MLP, NTN

<table>
<thead>
<tr>
<th>Model</th>
<th>Scoring Function</th>
<th>Relation Parameters</th>
<th>Space Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESCAL (Nickel et al., 2011)</td>
<td>$e_s^T W_r e_o$</td>
<td>$W_r \in \mathbb{R}^{d_e}$</td>
<td>$O(n_c d_e + n_r d_r^2)$</td>
</tr>
<tr>
<td>DistMult (Yang et al., 2015)</td>
<td>$\langle e_s, w_r, e_o \rangle$</td>
<td>$w_r \in \mathbb{R}^{d_e}$</td>
<td>$O(n_c d_e + n_r d_r)$</td>
</tr>
<tr>
<td>ComplEx (Trouillon et al., 2016)</td>
<td>Re($\langle e_s, w_r, \bar{e}_o \rangle$)</td>
<td>$w_r \in \mathbb{C}^{d_e}$</td>
<td>$O(n_c d_e + n_r d_r)$</td>
</tr>
<tr>
<td>ConvE (Dettmers et al., 2018)</td>
<td>$f(\text{vec}(f([e_s; w_r] * w)) W) e_o$</td>
<td>$w_r \in \mathbb{R}^{d_r}$</td>
<td>$O(n_c d_e + n_r d_r)$</td>
</tr>
<tr>
<td>HypER (Balažević et al., 2018)</td>
<td>$f(\text{vec}(e_s * \text{vec}^{-1}(w_r H)) W) e_o$</td>
<td>$w_r \in \mathbb{R}^{d_r}$</td>
<td>$O(n_c d_e + n_r d_r)$</td>
</tr>
<tr>
<td>SimplE (Kazemi &amp; Poole, 2018)</td>
<td>$\frac{1}{2}(\langle h_{e_s}, w_r, t_{e_o} \rangle + \langle h_{e_o}, w_{r-1}, t_{e_s} \rangle)$</td>
<td>$w_r \in \mathbb{R}^{d_e}$</td>
<td>$O(n_c d_e + n_r d_r)$</td>
</tr>
<tr>
<td>TuckER.</td>
<td>$W \times_1 e_s \times_2 w_r \times_3 e_o$</td>
<td>$w_r \in \mathbb{R}^{d_r}$</td>
<td>$O(n_c d_e + n_r d_r)$</td>
</tr>
</tbody>
</table>
Impact of Noise and Sparsity in KG embeddings

A large, unreliable training dataset may be better than an extremely sparse, high-quality one.

Pujara, Augustine, Getoor. Sparsity and Noise: Where Knowledge Graph Embeddings Fall Short. ACL 2017
https://www.github.com/ling/pujara-emnlp17
Link Prediction with Reinforcement Learning

No reward

- Leverage multi-hop KG query answering
- Use pre-trained model-based on-policy reinforcement learning
- New reward shaping and policy network with action dropout

Shaping Xi, Victoria Lin, Socher, Caiming Xiong. Multi-Hop Knowledge Graph Reasoning with Reward. EMNLP 2018
Link Prediction with Reinforcement Learning

- Leverage multi-hop KG query answering
- Use pre-trained model-based on-policy reinforcement learning
- New reward shaping and policy network with action dropout

Shaping Yi, Victoria Lin, Socher, Caiming Xiong. Multi-Hop Knowledge Graph Reasoning with Reward. EMNLP 2018
Joint Entity Linking with Deep Reinforcement Learning

WWW 2019, May 13-17, 2019, San Francisco, CA, USA Zheng Fang, Yanan Cao, Dongjie Zhang, Qian Li, Zhenyu Zhang, and Yanbing Liu

Figure 2: The overall structure of our RREL model. It contains three parts: Local Encoder, Global Encoder and Entity Selector. In this framework, \((V_{m_t}, V_{e_t}^k)\) denotes the concatenation of the mention context vector \(V_{m_t}\) and one candidate entity vector \(V_{e_t}^k\). The policy network selects one entity from the candidate set, and \(V_{a_t}\) denotes the concatenation of the mention context vector \(V_{m_t}\) and the selected entity vector \(V_{e_t}^k\). \(h_t\) represents the hidden status of \(V_{a_t}\), and it will be input into \(S_{t+1}\).
Identity Problem or Link Quality Problem?

**To assessing link quality:**
- Network topology and link properties
- Link type, content, and context
- Ontology axioms and ontology quality
- Provenance: source and extractor reliability
- Accessibility, reachability
- Information gain
- Task-dependent properties: e.g., in KG completion: path predicting power, path diversity (to avoid overfitting due to spurious paths)
Error Detection and Repairing

• **Error detection**
  Probabilistic techniques [Ruckhaus et al. 2014, Debattista et al., 2015, Li et al. 2015]

• **Value imputation**
  Statistics: SDType [Paulheim, Bizer, 2014],

• **Pattern enforcement**
  o Syntactic patterns (date formatting)
  o Semantic patterns (name/address)

• **Consistency checks and value update** to satisfy
  o A set of rules, constraints, FDs, CFDs, Denial Constraints (DCs), Matching Dependencies (MDs) with minimal number of changes
  o Integrity, Cardinality, Range or String-based constraints using W3C Shape Constraints Language (SHACL) and Shape Expressions Language (ShEX) [Rashid et al. 2019] see [http://github.com/AKSW/RDFUnit](http://github.com/AKSW/RDFUnit)
Consistency analysis in evolving KB

Hypothesis(H)

H1: Dynamics features from periodic data profiling can help to identify completeness issues.
H2: Learning models can be used to predict correct integrity constraints using the outputs of the data profiling as features.

<table>
<thead>
<tr>
<th>Learning Algorithm</th>
<th>Minimum Cardinality</th>
<th>Maximum Cardinality</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.9890</td>
<td>0.9574</td>
<td>0.9730</td>
</tr>
<tr>
<td>Least Squares SVM</td>
<td>0.9944</td>
<td>0.9468</td>
<td>0.9700</td>
</tr>
<tr>
<td>Multilayer Perceptron</td>
<td>0.9674</td>
<td>0.9468</td>
<td>0.9570</td>
</tr>
<tr>
<td>K-Nearest Neighbour</td>
<td>0.9511</td>
<td>0.9309</td>
<td>0.9409</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.9401</td>
<td>0.8351</td>
<td>0.8845</td>
</tr>
</tbody>
</table>

Rule discovery in KB


**RuleN**: [http://web.informatik.uni-mannheim.de/RuleN/](http://web.informatik.uni-mannheim.de/RuleN/)

**RUDIK**: [https://github.com/stefano-ortona/rudik](https://github.com/stefano-ortona/rudik)

---


Fine-Grained Evaluation: Rule-based vs embedding-based approaches

Concluding Remarks

• ML provides a principled framework and efficient tools for automating and optimizing many KG management tasks (e.g., extraction, population, completion, consistency checking)

• Paradox: ML for KG curation need high quality training data

• Hybrid approaches combining *Humans-in-the-loop, AutoML techniques and distant supervision* are promising for KG curation
Perspectives for ML-Based KG Curation

• Integrate the Human “in the Loop of ML-tools”
  – “Taskify” and minimize the amount of interactions with the users while, at the same time, maximize the potential “ML benefit” for KG management tasks

• Current efforts:
  Crowdsourcing, active learning, user-guided repair
  – Detecting LoD Quality issues via Crowdsourcing (DBpedia) [Acosta et al. 2016]
  – Data cleaning with oracle crowds [Bergman et al., SIGMOD’15]
  – User-guided repair of KB [Arioua, Bonifati, EDBT 2018]

• Direction:
  – Orchestration of Humans and ML-tools for KG curation
# Be inspired!

A Condensed View of ML-based curation solutions for structured data

<table>
<thead>
<tr>
<th>Repair System</th>
<th>ML Approach</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Febrrl</strong> [Churches et al., 2002]</td>
<td>HMM and MLE</td>
<td>Standardizing loosely structured texts (e.g., name/address) based on the probabilistic model learnt from training data</td>
</tr>
<tr>
<td><strong>SCARE</strong> [Yakout, Berti-Equille, Elmagarmid, SIGMOD’13]</td>
<td>Multiple ML models used to capture data dependencies across multiple data partitions</td>
<td>Find the candidate repair that maximizes the likelihood repair benefit under a cost threshold of the update</td>
</tr>
<tr>
<td><strong>Continuous Cleaning</strong> [Volkovs et al., ICDE’14]</td>
<td>Logistic classifiers</td>
<td>Learning from past user repair preferences to recommend next more accurate repairs</td>
</tr>
<tr>
<td><strong>Lens</strong> [Yang et al., VLDB’15]</td>
<td>Various ML models encoded in Domain Constraints</td>
<td>Declarative on-Demand ETL with prioritized curation tasks based on probabilistic query processing and PC-Tables</td>
</tr>
<tr>
<td><strong>HoloClean</strong> [Rekatsinas et al., VLDB 2017]</td>
<td>Probabilistic inference on factor graphs with SGD and Gibbs sampling</td>
<td>Mixing statistical and logical rules, DCs, MDs, etc. to infer candidate repairs in a scalable way with domain pruning and constraint relaxation</td>
</tr>
<tr>
<td><strong>BoostClean</strong> [Krishnan et al., 2017]</td>
<td></td>
<td>Mixing statistical and logical rules, domain constraints for detection and repair combinations to maximize the predictive accuracy over test data</td>
</tr>
<tr>
<td><strong>Learn2Clean</strong> [Berti-Equille, TheWebConf2019]</td>
<td>Reinforcement Learning</td>
<td>Learn from trial-and-errors the sequence of data preprocessing tasks that maximizes the quality of a given ML model</td>
</tr>
</tbody>
</table>

Poster #1293 on Wednesday!
Reinforcement learning for data cleaning
Learn2Clean: Optimizing the Sequence of Tasks for Data Preparation
[The Web Conference 2019]
Thanks!