Data Integration for Knowledge Graphs
Erhard Rahm
German AI Centers

5 new, permanent German AI centers (in addition to DFKI):

- Berlin (BIFOLD)
- Dortmund / Bonn (ML2R)
- Dresden / Leipzig (ScaDS.AI)
- München (MCML)
- Tübingen (tuebingen.ai)

www.humboldt-foundation.de
ScaDS.AI

• **ScaDS.AI**: Center for **Scalable Data Analytics** and **Artificial Intelligence**
  
  • extends previous Big Data center ScaDS Dresden/Leipzig (est. 2014)
  • since 2019: AI / Data Science center ScaDS.AI
  • **since July 2022: institutionally funded**
    • co-financed by BMBF and state of Saxony
## Research Areas

### Applied AI & Big Data
- Life Science & Medicine
- Environment & Earth Sciences
- Software Engineering
- Physics / Chemistry
- Engineering / Business

### AI Algorithms & Methods
- Understanding Language
- Methods and Hardware for Neuro-Inspired Computing
- Graph-based Artificial Intelligence
- Knowledge Representation & Engineering
- Scalable Visual Computing
- Federated, Efficient Learning
- Math Foundations & Statistical Learning

### Big Data Analytics & Engineering
- Big Data Analytics
- Open Data & Open Models
- Data Quality & Data Integration
- Responsible AI: Ethical and Societal Dimensions
- Architectures / Scalability / Security
Building up the center

• >150 employees
  • graduate school with about 100 Ph.D. students
  • service & transfer center with living labs in both Leipzig and Dresden

• 8+ **new AI/data science professorships**

• new **junior research groups** (5 so far)

• many additional 3rd-party projects and industry collaborations

• many events
AGENDA

- ScaDS.AI Dresden/Leipzig

- **Construction of Knowledge Graphs**
  - KG intro
  - requirements for KG construction
  - processing steps
  - comparison of existing approaches
  - open challenges

- Entity resolution / matching
  - ER intro
  - Entity clustering and incremental ER (Famer)
  - embedding-based matching of KGs

- Conclusions
Knowledge Graph Key Characteristics

A graph of data consisting of semantically described entities and relations of different types that are integrated from different sources.

- a graph (network) of "real world" entities
- high number of entity and relation types
- a formal semantic representation of things (e.g., using a KG ontology)
Importance of Knowledge Graphs

- background knowledge
- semantic search
- QA
- recommender systems

- ML support
  - training data
  - Classification
  - improved explainability

https://lod-cloud.net/
Marburg

Marburg is a German town north of Frankfurt. It's home to Philipps University, founded in 1527. The Alstadt, or old town, includes half-timbered houses and the hilltop Landgrafenschloss, a castle with exhibits on sacred art and regional history. Bars and cafes line Marktplatz square and the narrow streets surrounding it. The 13th-century, Gothic-style St. Elizabeth's Church holds a shrine with the saint's remains. — Google

Weather: 62°F (17°C), Wind SW at 10 mph (16 km/h), 58%
Humidity: 59% More on weather.com

Local time: Wednesday 4:11 PM
District: Marburg-Biedenkopf
Highest elevation: 412 m (1,352 ft)
Postal codes: 35001-35043

Cost of living
Cost of living in marburg germany

History
Marburg germany history

Events
Marburg germany events

Closest airport
Closest airport to marburg germany

Is Marburg worth a visit?

Located in Hessen, Germany, Marburg is home to an impressive selection of attractions and experiences, making it well worth a visit. Located in Hessen, Germany, Marburg is home to an impressive selection of attractions and experiences, making it well worth a visit. Wed. Thur.

https://www.trip.com/destination/marburg-27368
Marburg Travel Guide 2023 - Things to Do, What To Eat & Tips | Trip.com
Example: Product Knowledge Graph

Graph Applications
- Querying
- Graph Mining
- Embedding Generation
- Recommendation
- Search, QA, Conversation

Graph Construction
- Knowledge Cleaning
  - Schema Mapping
  - Entity Resolution
  - Knowledge Cleaning

Knowledge Collection
- Ontology
- Ingestion
- Web Extraction
- Catalog Extraction

from: Dong. KDD2018
Knowledge Graph Construction

- unstructured (TEXT) or multimodal data (audio, images, videos)
- semi-structured (e.g., JSON, CSV)
- structured (RDB, KGs)

Wikidata knowledge graph example using SPARQL by Fuzheado is licensed under CC BY 4.0 SA
Construction of Knowledge Graphs: State and Challenges

Marvin Hofer b,a, Daniel Obraczka b, Alieh Saeedi a,b, Hanna Köpcke a and Erhard Rahm a,b

a Dept. of Computer Science, Leipzig University, Germany
b Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI) Dresden/Leipzig, Germany

Abstract. With knowledge graphs (KGs) at the center of numerous applications such as recommender systems and question answering, the need for generalized pipelines to construct and continuously update such KGs is increasing. While the individual steps that are necessary to create KGs from unstructured (e.g. text) and structured data sources (e.g. databases) are mostly well-researched for their one-shot execution, their adoption for incremental KG updates and the interplay of the individual steps have hardly been investigated in a systematic manner so far. In this work, we first discuss the main graph models for KGs and introduce the major requirement for future KG construction pipelines. Next, we provide an overview of the necessary steps to build high-quality KGs, including cross-cutting topics such as metadata management, ontology development, and quality assurance. We then evaluate the state of the art of KG construction w.r.t the introduced requirements for specific popular KGs as well as some recent tools and strategies for KG construction. Finally, we identify areas in need of further research and improvement.

Keywords: Knowledge Graph, Data Integration, Data Science
Requirements for KG construction

- **Input Data Requirements**
  - support for many, large and heterogenous data sources
  - techniques for data acquisition, knowledge extraction, entity resolution/fusion

- **Support for Incremental KG updates**
  - process new input data in batches or continuously in a streaming manner
  - series of batch-created KG versions vs. incremental updates of changes/new sources
  - tradeoffs in simplicity vs. scalability /freshness

- **Pipeline and Tools Requirements**
  - tool support needed to simplify KG construction (creation of application-specific pipelines)
  - utilize existing, independently developed tools
  - simplified configuration of individual steps
  - support for debugging and tuning

- **Quality Assurance**
  - ensure high data quality in individual pipeline steps and in resulting KG
Pipeline Blueprint

Un-, Semi- or Structured Input Sources (+ KG and Configs)

Task Pipeline and MetaData Management
- Ontology Management
- Knowledge Extraction
- Entity Resolution & Fusion

Configurations (Schemas, Mappings)

Data Repository
- Ingestion & Updates
- Data Versions
- Data Provenance

Knowledge Completion

Integrated Knowledge Graph Versions
- KG_n
- KG_1

Data Management Layer
(Cleaning, Mapping)
Overview of KG Construction Tasks

- **Initial KG construction**: manual crowdsourcing, sampling existing KG
- **Data preprocessing**: data acquisition, data cleaning and transformation
- **Metadata management**: persistence, access, versioning, provenance
- **Ontology development**: creation, evolution, integration
- **Knowledge extraction**: entity recognition, linking, relation extraction
- **Entity resolution**: entity matching, clustering, data fusion
- **Quality assurance**: quality assessment, repair, debugging
- **Knowledge completion**: type-, link prediction, enrichment, polishing

*cross-cutting and special tasks*
Knowledge Extraction

- bringing unstructured or semi-structured data to structured, machine-readable information
- subtasks: **Named-Entity Recognition (NER), Entity Linking (EL), and Relation Extraction (RE)**
- multi-modal KE: visual relation extraction from images

Diagram:

- Richard David James returned in 2014 with the album Syro.
- Named Entity Recognition
- Disambiguation
- Relation Extraction & Linking
- dbpedia:Syro rdf:type dbpedia:Album .
Quality Assurance

- high KG quality crucial for credibility and usability

- subtasks: **quality evaluation** (identifying issues) and **quality improvement** (fixing issues) / **KG completion**

- Quality evaluation
  - dimensions: accuracy, consistency, timeliness, completeness, trustworthiness, availability
  - manual checks (experts, crowd-sourcing), statistical analysis, semantic reasoning, comparison with external sources

- Quality improvement
  - Error correction, data cleaning, entity resolution and fusion
  - ontology evolution

- **Knowledge completion:** improve KG by new nodes, relations, properties
  - type completion: Assigning types to nodes lacking type information using node classification, logical reasoning, or statistical approaches.
  - link prediction: Identifying missing relations in KG, with techniques like distant supervision, embedding-based methods, or Graph Neural Networks.
  - data enrichment: add entity information from external knowledge bases, e.g. using persistent identifiers (ISBN, DOIs, ORCIDs ...)

ScaDS.AI
Dresden Leipzig
Exemplary Selection and Comparison

- Investigation of 23 specific KGs/construction approaches and toolsets
  - 3 closed KGs: Google, Diffbot, Amazon
  - 3 manually curated KGs: Freebase, Wikidata, ORKG
  - 10 open KGs: DBPedia, DBPedia-live, YAGO, NELL, ArtistKG, CovidKG, ...
- 7 toolsets for KG construction: FlexiFusion, dstlr, XI, Autoknow, HKGB, SLOGERT; Saga
- selection based on relevance (popularity), novelty, existing paper/documentation, with multiple versions
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Year</th>
<th>Domain</th>
<th>Srcs.</th>
<th>Model</th>
<th>Entities</th>
<th>Relations</th>
<th>Types</th>
<th>R-Types</th>
<th>Vers.</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closed KG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diffbot.com</td>
<td>2019</td>
<td>Cross</td>
<td>&gt;&gt;&gt;1</td>
<td>RDF</td>
<td>5.9B</td>
<td>&gt;1T</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Amazon PG [196]</td>
<td>2020</td>
<td>Products</td>
<td>&gt;1</td>
<td>Custom</td>
<td>30M</td>
<td>1B</td>
<td>19K</td>
<td>1K</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Open Access KG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBpedia [198]</td>
<td>2007</td>
<td>Cross,MLang</td>
<td>140</td>
<td>RDF</td>
<td>50M</td>
<td>21B</td>
<td>1.3K</td>
<td>55K</td>
<td>&gt;20</td>
<td>2023</td>
</tr>
<tr>
<td>NELL [201]</td>
<td>2010</td>
<td>Cross</td>
<td>≥1</td>
<td>Custom,RDF</td>
<td>2M</td>
<td>2.8M</td>
<td>1.2K</td>
<td>834</td>
<td>&gt;1100</td>
<td>2018</td>
</tr>
<tr>
<td>*Wikidata [202]</td>
<td>2012</td>
<td>Cross,MLang</td>
<td>&gt;&gt;&gt;1</td>
<td>RDB/RDF</td>
<td>100M</td>
<td>14B</td>
<td>300K</td>
<td>10.3K</td>
<td>&gt;100</td>
<td>2023</td>
</tr>
<tr>
<td>DBpedia-EN Live [203]</td>
<td>2012</td>
<td>Cross</td>
<td>1</td>
<td>RDF</td>
<td>7.6M</td>
<td>1.1B</td>
<td>800</td>
<td>1.3K</td>
<td>&gt;&gt;&gt;1</td>
<td>2023</td>
</tr>
<tr>
<td>Artist-KG [204]</td>
<td>2016</td>
<td>Artists</td>
<td>4</td>
<td>Custom</td>
<td>161K</td>
<td>15M</td>
<td>&gt;1</td>
<td>18</td>
<td>1</td>
<td>2016</td>
</tr>
<tr>
<td>*ORKG [205]</td>
<td>2019</td>
<td>Research</td>
<td>&gt;&gt;&gt;1</td>
<td>RDF</td>
<td>130K</td>
<td>870K</td>
<td>1.3K</td>
<td>6.3K</td>
<td>&gt;1</td>
<td>2023</td>
</tr>
<tr>
<td>AI-KG [206]</td>
<td>2020</td>
<td>AI Science</td>
<td>3</td>
<td>RDF</td>
<td>820K</td>
<td>1.2M</td>
<td>5</td>
<td>27</td>
<td>2</td>
<td>2020</td>
</tr>
<tr>
<td>CovidGraph [207]</td>
<td>2020</td>
<td>COVID-19</td>
<td>17</td>
<td>PGM</td>
<td>36M</td>
<td>59M</td>
<td>128</td>
<td>171</td>
<td>&gt;1</td>
<td>2020</td>
</tr>
<tr>
<td>DRKG [208]</td>
<td>2020</td>
<td>BioMedicine</td>
<td>&gt;7</td>
<td>CSV</td>
<td>97K</td>
<td>5.8M</td>
<td>17</td>
<td>107</td>
<td>1</td>
<td>2020</td>
</tr>
<tr>
<td>VisualSem [209]</td>
<td>2020</td>
<td>Cross,MLang</td>
<td>2</td>
<td>Custom</td>
<td>90k</td>
<td>1.5M</td>
<td>(49K)</td>
<td>13</td>
<td>2</td>
<td>2020</td>
</tr>
<tr>
<td>WorldKG [210]</td>
<td>2021</td>
<td>Geographic</td>
<td>1</td>
<td>RDF</td>
<td>113M</td>
<td>829M</td>
<td>1176</td>
<td>1820</td>
<td>1</td>
<td>2021</td>
</tr>
</tbody>
</table>

* manually curated
<table>
<thead>
<tr>
<th>Name of System</th>
<th>System Version/Year</th>
<th>Open Implementation</th>
<th>Incremental Integration</th>
<th>Consumed Data</th>
<th>(Meta)Data</th>
<th>Performed Construction Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Deep Provenance</td>
</tr>
<tr>
<td><strong>Dataset Specific</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBpedia</td>
<td>2019</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>YAGO4</td>
<td>2020</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DBpedia-Live</td>
<td>2012</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NELL</td>
<td>2018</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Artist-KG</td>
<td>2016</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>AI-KG</td>
<td>2020</td>
<td>✓</td>
<td>?</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CovidGraph</td>
<td>2020</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DRKG</td>
<td>2020</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>VisualSem</td>
<td>2020</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>WorldKG</td>
<td>2021</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Toolset/Strategy</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FlexiFusion [90]</td>
<td>2019</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dstir [137]</td>
<td>2019</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>XI [50]</td>
<td>2020</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>AutoKnow [196]</td>
<td>2020</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HKGB [211]</td>
<td>2020</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SLOGERT [212]</td>
<td>2021</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SAGA [47]</td>
<td>2022</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- ✓ supported/provides
- ○ simple/manual
- ▪ sophisticated/semi-automatic
- ? unclear implementation
<table>
<thead>
<tr>
<th>Name of System</th>
<th>System Version/Year</th>
<th>Open Implementation</th>
<th>Incremental Integration</th>
<th>Consumed Data</th>
<th>(Meta)Data</th>
<th>Performed Construction Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dataset Specific</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBpedia</td>
<td>2019</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>YAGO4</td>
<td>2020</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>DBpedia-Live</td>
<td>2012</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>NELL</td>
<td>2018</td>
<td>•</td>
<td>✓</td>
<td>✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>Artist-KG</td>
<td>2016</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>AI-KG</td>
<td>2020</td>
<td>?</td>
<td>✓</td>
<td>✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>CovidGraph</td>
<td>2020</td>
<td>✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>DRKG</td>
<td>2020</td>
<td>✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>VisualSem</td>
<td>2020</td>
<td>✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>WorldKG</td>
<td>2021</td>
<td>✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
</tbody>
</table>

| Toolset/Strategy      |                     |                     |                         |                                                                                   |                   |                             |
| FlexiFusion [90]      | 2019                | ?                   | ✓ ✓ ✓                   | ✓ ✓ ✓                                                                                  | ✓ ✓ ✓            | ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ |
| dstr [137]            | 2019                | ?                   | ✓ ✓ ✓                   | ✓ ✓ ✓                                                                                  | ✓ ✓ ✓            | ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ |
| XI [50]               | 2020                | ?                   | ✓ ✓ ✓                   | ✓ ✓ ✓                                                                                  | ✓ ✓ ✓            | ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ |
| AutoKnow [196]        | 2020                | ✓                   | ✓ ✓ ✓                   | ✓ ✓ ✓                                                                                  | ✓ ✓ ✓            | ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ |
| HKGB [211]            | 2020                | ✓                   | ✓ ✓ ✓                   | ✓ ✓ ✓                                                                                  | ✓ ✓ ✓            | ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ |
| SLOGERT [212]        | 2021                | ✓                   | ✓ ✓ ✓                   | ✓ ✓ ✓                                                                                  | ✓ ✓ ✓            | ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ |
| SAGA [47]            | 2022                | •                   | ✓ ✓ ✓                   | ✓ ✓ ✓                                                                                  | ✓ ✓ ✓            | ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ |

- ✓ supported/provides
- • sophisticated/semi-automatic
- ○ simple/manual
- ? unclear implementation

simple matching, no fusion
**SAGA tool** (Apple, Ilyas et al., Sigmod 2022)
Open challenges in KG construction

- better support for **incremental KG construction**
  - batch-like KG re-creation has limited scalability and out-of-date information
  - more complex: change detection in sources and incremental pipeline
- **lack of open tools** for KG construction
- toolset for defining different KG construction pipelines with different implementations for certain tasks (extensible, modular approach needed)
- more comprehensive approaches needed for **metadata management** and **KG quality assurance**
- **evaluation of KG construction approaches**
  - so far only benchmarks for single tasks (extraction, matching, completion)
  - not sufficient to evaluate/compare different end-to-end construction approaches
- **use of Large Language Models (LLMs)** for KG construction
AGENDA

- ScaDS.AI Dresden/Leipzig
- Construction of Knowledge Graphs
  - KG intro
  - requirements for KG construction
  - processing steps
  - comparison of existing approaches
  - open challenges

- **Entity resolution / matching**
  - ER intro
  - entity clustering and incremental ER (Famer)
  - embedding-based matching of KGs

- Conclusions
DATA MATCHING / ENTITY RESOLUTION

- Identification of semantically equivalent objects
- within one data source or between different sources

Fujifilm FinePix S6800

- **manufacturer**: Fujifilm
- **resolution**: 16.2 MP
- **model**: FinePix S6800
- **zoom**: 30x
- **weight**: 0.43 kg

- **brand**: Fujifilm
- **model**: Point & Shoot S6800
- **weight**: 430 gram
- **color**: black

- **type**: Point & Shoot
DUPLICATE PUBLICATION ENTRIES

Data cleaning: Problems and current approaches
Cited by: 2790 Related articles: All 29 versions

Data Cleaning: Problems & Current Approaches
D Hang Hai, R Erhard - IEEE bulletin of the technical committee on Data ..., 2000
Cited by: 8 Related articles

Problems and Current Approaches
E Rahm, DC Do HH - IEEE Bulletin on Data Engineering -2000 -23 (4), 2015
Cited by: 7 Related articles

E Rahm, H Do - 2000
Cited by: 7 Related articles

Data engineering—Special issue on data cleaning
E Rahm, HH Do - Data Engineering, 2000
Cited by: 5 Related articles

Data Cleaning: Problems and Current Approaches, IEEE Techn *
E Rahm, HH Do - Bulletin on Data Engineering, 2000
Cited by: 5 Related articles

E Rahm, H Do - 2000
Cited by: 5 Related articles

Do, H. 2000. Data cleaning: Problems and current approaches *
E Rahm, HAI HONG - IEEE Data Engineering Bulletin
Cited by: 5 Related articles
ER CHALLENGES

- **Scalability**
  - large data volume or/and many sources
  - need to reduce search space (e.g. with blocking) + parallel processing

- **High match quality**
  - low quality input data (unstructured, semi-structured sources)
  - needs effective combination of several techniques
  - use of supervised ML approaches
  - use of entity embeddings

- **Support for evolution and change**
  - addition of new sources and new entities without having to integrate everything again
  - incremental / dynamic vs batch / static ER
ENTITY RESOLUTION WORKFLOW

- mostly only 1 or 2 sources
- \( n \geq 2 \): duplicate-free (clean) sources or not
  - clean sources: at most one entity per cluster (cluster sizes \( \leq n \))
BLOCKING & FILTERING

- naïve: pairwise matching of all entities
  - quadratic complexity, not scalable
  - strong need to reduce match search space

- Blocking
  - group similar objects within blocks / partitions
  - only compare entities of the same block
  - many variations: Standard Blocking, LSH, Sorted Neighborhood, ...

- Filtering
  - typically applied for similarity joins with fixed threshold $t$: $\text{sim} (e_1, e_2) \geq t$
  - utilizes characteristics of similarity function, e.g., for string similarity
  - for embeddings: only consider nearest neighbors
Fig. 3. The genealogy trees of nonlearning (a) schema-aware and (b) schema-agnostic Block Building techniques. Hybrid, hash-, and sort-based methods are marked in blue, black, and red, respectively.
MATCHING

- combined use of several similarity values
  - attribute similarities, e.g. using numeric or string similarity measures
  - context-based matchers

- general match rules with multiple similarities
  - e.g. pubs match if title sim. ≥ 0.9 & author sim. > 0.4

- learned/supervised match classification models
  - need suitable training data
FAMER TOOL

- **FAst Multi-source Entity Resolution System**
  - scalable linking & clustering for many sources

**Input**

- Source A
- Source B
- Source C
- Source D
- Source E

**Linking:** Similarity Graph

**Clustering**
FAMER BATCH PIPELINE

Data Sources

source 1

source 2

source k

LINKING

Blocking
Pair-wise Comparison
Match Classification

CLUSTERING

Cluster Set
CLIP APPROACH (ESWC BEST RESEARCH PAPER)

- optimized for clean sources
- CLIP (CLustering based on Link Priority) uses **link strength**
  - **strong**: maximum link from both ends
  - **normal**: maximum link from one end
  - **weak**: maximum link from no end
- CLIP
  - ignores weak links
  - focusses on strong links
  - also considers normal links
EVALUATION: GEO. DATASET

Precision

Recall

F-Measure

InputGraph  ■  ConCom  ▲  CC Pivot  ▼  Center  □  M Center  ◊  Star1  ◊  Star2  ▪  CLIP
MULTI-SOURCE CLEAN/DIRTY CLUSTERING

- previous assumption: data sources are duplicate-free
- more realistic assumption: some sources are dirty
  - solution: first deduplicate dirty sources
  - problem: requires immense effort and perhaps not completely successful
- solution: MSCD approaches
  - approaches that can deal with dirty sources
  - only a fraction (possibly 0%) of sources have to be clean
  - goal: achieve better match quality than general clustering scheme while avoiding limitation of requiring duplicate-free sources
  - most promising: hierarchical agglomerative clustering (HAC)
MSCD-HAC

- modify Hierarchical Agglomerative Clustering -> MSCD-HAC

- iterative approach
  - initially each entity forms a cluster
  - continuously determine most similar pair of clusters \((c_i, c_j)\) as long as minimal merge sim. threshold is exceeded. Merge clusters \(c_i, c_j\) only when
    - they are Reciprocal Nearest Neighbours (RNN), i.e. \(\text{NN}(c_j) = c_i\) and \(\text{NN}(c_i) = c_j\)
    - observe that at most one entity of a clean source in a cluster

- 3 approaches to determine cluster similarity \(\text{sim} (c_i, c_j)\)
  - Single linkage (S-LINK): \(\text{sim} c_i, c_j = \max \{\text{sim}(e_m, e_n)\}\)
  - Complete linkage (C-LINK): \(\text{sim} c_i, c_j = \min \{\text{sim}(e_m, e_n)\}\)
  - Average linkage (A-LINK): \(\text{sim} c_i, c_j = \text{avg} \{\text{sim}(e_m, e_n)\}\)
F-MEASURE: CAMERA DATASET

match threshold = merge threshold ($\theta$)

- DS-C0
  - 0% clean
  - MSCD A-LINK
  - MSCD C-LINK

- DS-C50
  - 50% clean
  - MSCD S-LINK

- DS-C100
  - 100% clean
  - MSCD S-LINK
  - CLIP
MOTIVATION

- static one-time matching and clustering insufficient
- need for incremental approaches
  - data sources change over time
  - new relevant data sources are added continuously
- expensive re-computation of similarity graph/clusters to be avoided
- order in which new entities are added should have minimal impact
  - need to repair wrong clusters
FAMER INCREMENTAL PIPELINE
FAMER N-DEPTH RECLUSTERING

- requires to keep similarity graphs for clustered entities
- recluster new entities in $G_{\text{new}}$ with their neighbors
  - can repair old cluster decisions
  - limits amount of reclustering for efficiency
  - reduce dependence on order of entity additions
- evaluation results
  - incremental approaches are much faster
    and similarly effective than batch ER
  - quality of nDR does not depend on the order in which new entities are added
ENTITY RESOLUTION ON KNOWLEDGE GRAPHS

- similar ER challenges as discussed
  - large KGs (e.g., 100 million entities in Wikidata)
- ER for many interrelated entity types needed
  - standard ER assumes only 1 entity type

- **Key idea**: map entities of input KGs into embedding space and determine matches based on nearest neighborhood
  - word embeddings for properties/attribute values
  - graph embeddings to consider neighboring entities in KG
KNOWLEDGE GRAPH EMBEDDINGS (KGE)

- transform entities into a dense vector so that
  - similar entities close in the embedding space
  - relational information is retained
- many possible approaches
  - translational KGEs for triples <h,r,t>
    (e.g. MultiKE, BootEA)
  - Graph Neural Network approaches
    (e.g. RDGCN, CG-MuAlign)
    based on aggregated entity neighborhood in KG
EAGER: EMBEDDING-ASSISTED ENTITY RESOLUTION FOR KG

Obraczka, Schuchart, and Rahm, "Embedding-Assisted Entity Resolution for Knowledge Graphs", 2021
EXPERIMENTAL EVALUATION

- 16 alignment tasks
  - KG subsets from DBpedia, Wikidata, YAGO
  - different densities, sizes and even cross-lingual settings
- 3 KG embedding approaches (BootEA, MultiKGE, RDGCN)
- comparison of 3 approaches
  - OnlyEmb – only graph embeddings are used
  - OnlySim: only attribute similarities are used
  - SimAndEmb: use both
Results for 100K datasets (using MLP as classifier)
PROBLEMS WITH EMBEDDINGS

- Problems with runtime and quality for larger and more diverse KGs
- Blocking approaches not applicable to speed-up matching
- Exact nearest-neighbor algorithms become slow
  - Need to apply faster approximate nearest neighbor (ANN) algorithms, e.g. Annoy, Faiss...
- But ANN algorithms lose some matches (reduced recall)
  - Embeddings are relatively high-dimensional (> 200)
  - "Hubness" of embedded entities
HUBNESS REDUCES ALIGNMENT QUALITY

with increasing dimensionality:

- few points are nearest neighbors (NN) of many points
- many points are NN of no points

⇒ hubness negatively affects alignment quality
kiez

open-source python library (github.com/dobraczka/kiez) for hubness-reduced nearest neighbor search (for entity alignment with knowledge graph embeddings)

Obreczka and Rahm, “An Evaluation of Hubness Reduction Methods for Entity Alignment with Knowledge Graph Embeddings”, 2021
kiez

Open-source python library (github.com/dobraczka/kiez) for hubness-reduced nearest neighbor search (for entity alignment (with knowledge graph embeddings))

(Approximate) Nearest Neighbor Method:
- Sci-kit learn  
  - Pedregosa et al., 2011
- BallTree  
  - Omohundro, 1989
- KDTree  
  - Bentley, 1975
- Bruteforce
- NMSLIB: HNSW  
  - Malkov, 2018
- NGT  
  - Iwasaki, 2016
- Annoy  
  - (github.com/spotify/annoy)
- Faiss  
  - Johnson, Douze, and Jégou, 2017

Hubness reduction methods:
- Local Scaling  
  - Schnitzer et al., 2012
- NICDM  
  - Schnitzer et al., 2012
- CSLS  
  - Lampl et al., 2018
- Mutual Proximity  
  - Schnitzer et al., 2012
- DisSimLocal  
  - Hara et al., 2016
EVALUATION RESULTS

- hubness reduction improves alignment results

- using ANN algorithms (Faiss) with hubness reduction approach (NICDM) gives improvements at virtually no cost w.r.t speed

⇒ hubness reduction largely offsets decrease in alignment quality when using approximate nearest neighbor algorithm while still retaining speed advantage
FUTURE DIRECTIONS FOR KGE-BASED METHODS

- more realistic evaluations\(^1\)
  - differently sized KGs, not only 1:1 matches, ...

- better scalability of KGE-based methods
  - blocking-like approaches not yet explored

- dealing with unseen entities is almost unexplored\(^2\)

- unsupervised KGE approaches, e.g. for clustering

---

\(^1\) Leone et al., "A Critical Re-evaluation of Neural Methods for Entity Alignment", 2022
\(^2\) Wang et. al.,"Facing Changes: Continual Entity Alignment for Growing Knowledge Graphs", 2022
SUMMARY

- largely automatic creation/refinement of large knowledge graphs is still difficult
  - open toolsets needed supporting all major steps with easy configuration
  - better approaches needed for incremental updates, quality assurance, ontology evolution, multi-modal KGs ...
  - holistically evaluating KG construction approaches is challenging

- **Entity resolution**
  - huge amount of previous work mostly on structured and static data for single kind of entities
  - need for incremental approaches for KGs with many entity types
  - use of KG embeddings promising but with need for improvements
References Knowledge Graphs

- Dong et al. (2020). AutoKnow: Self-Driving Knowledge Collection for Products of Thousands of Types. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.
References

- Wilkinson et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data
References Data Integration / Entity resolution

- L. Dong: Challenges and Innovations in Building a Product Knowledge Graph. Tutorial, KDD 2018
- A. Gruenheid et al.: Incremental record linkage. VLDB 2014
- O. Hassanzadeh et al.: Clustering for Duplicate Detection. VLDB 2009
References

- S. Lerm, A. Saeedi, E. Rahm: Extended Affinity Propagation Clustering for Multi-source Entity Resolution. BTW 2021
- D. Obraczka, A. Saeedi, A. E. Rahm, E.: Knowledge Graph Completion with FAMER. Proc. KDD DI2KG, 2019
- A. Saeedi, L. David, E. Rahm, E: Matching Entities from Multiple Sources with Hierarchical Agglomerative Clustering. KEOD 2021
References

- A. Saeedi, M. Nentwig, E. Peukert, E. Rahm: Scalable matching and clustering of entities with FAMER. CSIM Quarterly 2018
- A. Saeedi, E. Peukert, E. Rahm: Using Link Features for Entity Clustering in Knowledge Graphs. ESWC 2018
- A. Saeedi, E. Peukert, E. Rahm: Incremental Multi-source Entity Resolution for Knowledge Graph Completion. ESWC 2020
- J. Shao, Q.; Wang, A. Wijesinghe, E. Rahm: ERGAN: Generative Adversarial Networks for Entity Resolution. ICDM 2020
- M. Wilke, E. Rahm: Towards Multi-modal Entity Resolution for Product Matching. GVDB 2021
- Zhang, Q., Sun, Z., Hu, W., Chen, M., Guo, L., & Qu, Y. (2019). Multi-view knowledge graph embedding for entity alignment. IJCAI International Joint Conference on Artificial Intelligence, 2019