Spread the good around! Information Propagation in Schema Matching and Entity Resolution for Heterogeneous Data

(Experience Report)
DI2KG 2020 Challenge Winner Paper

Gabriel Campero Durand, Anshu Daur, Vinayak Kumar, Shivalika Suman, Altaf Mohammed Aftab, Sajad Karim, Prafulla Diwesh, Chinmaya Hegde, Disha Setlur, Syed Md Ismail, David Broneske, Gunter Saake
University of Magdeburg

ABSTRACT

In this short paper we describe the experience from our entries to the Entity Resolution (ER) and Schema Matching (SM) challenges of the Second DI2KG workshop. Altogether we study four solutions, two domain-specific and two based on machine learning (ML). Concerning ER, we find that through ample data cleaning/extraction, simple matching rules can already achieve a high f1 score (0.921). However, we note the limited generalization power of such kind of solutions. For ML-ER, by reducing data cleaning/extraction, generic ML models resulted unsuccessful out of the box; but by increasing it, models resulted redundant compared to simple rules. For SM, we report less competitive f1 scores, establishing the need for more appropriate methods than those attempted. Based on our experience we confirm the importance of automating data cleaning/extraction as a goal towards general data integration methods that would be more portable across datasets. We venture that for highly heterogeneous schemas, a promising approach could be to evolve collective integration with ML & graph-based methods, incorporating strategies based on information propagation.

CCS CONCEPTS

• **Information systems → Entity resolution**; *Data cleaning*; **Mediators and data integration**.

KEYWORDS

Entity Resolution, Schema Matching, Data Extraction, Data Cleaning

1 INTRODUCTION

Data integration is a foundational activity that can make a compelling difference in the workflow and bottom-line performance of data-driven businesses. Either when contributing towards a 360 degree user understanding that can translate to personalized services, when helping disease specialists track the latest global information on viruses from heterogeneously presented official resources, or when improving the knowledge that an e-commerce platform has on the products of its sellers, data integration plays (or can play) a definite vital role in everyday missions.

For AI pipelines, data integration is an early step from the data conditioning/wrangling stage (i.e., covering standardization and augmentation). In this context, integration is merely one operation from a larger process to improve data readiness, including data discovery, imputation of missing values, among others [8]. On more traditional data management cases, data integration can be scoped to be a series of tasks that provide users with a consolidated interface to utilize heterogeneous data sources [10]. Some data integration tasks include *entity resolution* (ER, i.e., determining pairs of records that refer to the same entity), *schema matching* (SM, which could be a sub-task of ER and refers to finding correspondences between elements of different schemas, possibly matching them to a mediated schema) and *data fusion* (i.e., combining all the data from different entity-resolved records to a single “golden record” representation, using a target mediated schema) [11].

There’s a lot of diversity in dataset characteristics and integration application scenarios. This poses many challenges for the tasks, propitiating today’s ecosystem of numerous specialized offers plus a few holistic systems. The generational evolution of research in ER, as presented by Papadakis et al [16], illustrates some of such varied integration application scenarios, and some of the tools developed for them. Focusing on the specific task of ER, authors observe four generations of tools, with early developments (1st and 2nd Gen) assuming a relational context with clean (& homogeneous) schema designs that are known up-front, additionally they might include the expectation of big data volumes, requiring large-scale processing (2nd Gen). On the other hand, more recent approaches either strive to address the inherently great schema heterogeneity of Web data (3rd Gen), progressive/streaming ER scenarios (4th Gen) [15], or they return to the case of homogeneous schemas (as studied for 1st Gen tools), but leveraging the possibilities of semantic matching over noisy data, with deep learning.

The recently proposed DI2KG benchmarks seek to foster a cross-disciplinary community for building the next generation of data integration and knowledge graph construction tools. These benchmarks cover challenging datasets for every data integration task. The availability of the benchmarks should help standardize comparisons of proposed methods, and further the understanding of trade-offs between dataset-tailored and more generic techniques.

In this paper we describe four relatively simple solutions we developed for the ER and SM tasks of the DI2KG benchmark. The DI2KG challenge provides datasets of product descriptions from e-commerce services for camera, monitor and notebook data. For our study we use the monitor dataset. It consists of 16,662 JSON files from 26 sources, with a substantial amount of schema heterogeneity,

DI2KG 2020, August 31, Tokyo, Japan. Copyright ©2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

http://di2kg.inf.uniroma3.it/

and noise (in the form of ill-parsed data, non-product data, multilingual text, and others). Hence, the dataset comprises of a mix of challenges not trivially mapped to a single approach from the literature. In order to understand better how to solve them, we pursued for each task relatively simple domain-specific and ML-based variants. To summarize, we contribute with:

- A dataset-tailored solution establishing that for the ER task, abundant data cleaning/extraction provides success with trivial matching.
- Dataset-tailored and ML-based SM solutions, relying on instance-level information, as baselines for improvement.

The remainder of this paper is organized in three sections, covering a concise background relevant to our proposed solutions (Sec. 2), the description of our developed tools, with their corresponding evaluation results (Sec. 3), and a closing summary with suggestions for further research (Sec. 4)

2 BACKGROUND

Entity Resolution: ER has a history that spans almost 5 decades [6], with a trend for applying supervised learning, and specially deep learning [20], growing in recent years [16]. Related work on DeepER [5] reports good results over schema-homogeneous product data, by using locality sensitive hashing, pre-trained models for generating entity embeddings, and neural models for similarity comparisons & embedding composition. Addressing similar datasets, Mudgal et al., with DeepMatcher, [14] report comprehensive evaluations on variations for the steps of attribute embedding, similarity representation (incl. summarization/composition), and comparison; showing a competitive edge on structured textual and dirty data, over a state-of-the-art tool. More recently Brunner and Stockinger [2] employed transformers (BERT, and others) on a scenario similar to DeepMatcher. To the best of our knowledge deep learning methods using information from more than one entity at inference time are uncommon; however some early studies report promising results over proprietary datasets[12]. Parallel to the work in deep learning, tools such as JedAI[17], FAMER[15] or Magellan[4] support highly configurable human-in-the-loop large-scale or cloud-based ER.

Schema Matching: Schema matching has traditionally been researched in the context of relational data, with authors taking approaches based on structural similarity (e.g. name similarity, or concerning PK:FK relationships), instance-based similarity (i.e. considering the distribution of values for the attributes being compared), or hybrids [3, 13, 21]. Bernstein et al. survey a large list of techniques for SM in use by 2011[1]. More recently an approach called Seeping Semantics [7] has employed semantic (average cosine similarity of word embedding vectors) and syntactic similarity (names and instance values) for matching across datasets. Recent work also addresses the related task of creating a mediated schema through decision-tree rules that can be presented to end-users[9].

	Explicit brand naming	Alternative names & cleaning
# of brands	202	321
# of items with brands assigned	8,015	16,653

Table 1: Effectiveness of steps for brand extraction

3 PROPOSED SOLUTIONS FOR THE DI2KG CHALLENGE & EVALUATION

3.1 Entity Resolution

Domain-Knowledge Approach: Based on some exploratory analysis we identified that *brands* and *models* were likely to be highly informative fields to form non-overlapping groups for the pair-wise comparisons of ER. Hence, we started by tackling the problems of brand and model assignment. For *brand assignment* we found that already 48% of our data (8015 items) had explicit information under 8 attributes: brand, brand name, manufacturer, product name, product line, label, publisher, and studio (see Table 1). Subsequent to a first assignment, with standardization in the naming, we found 202 brands with only a few of them (17) representing 83% (6,597 items) of the data with brand assignments. With the knowledge of brand names, and enhancing this with information on sub-brands (e.g. Asus and Republic of Gamers) and short-forms/aliases (e.g. HP and Hewlett-Packard), we could now seek for such names in the page titles of the products. Through this method we could immediately assign brands to 16,653 items (incl. unbranded/generic). Following ample rule-based cleaning of our assignments (to cover edge-cases identified), the brands were narrowed down to 321, with 22 brands covering 15,521 of the products, and only 149 items being considered unbranded/generic. During the brand cleaning process, (establishing that items considered generic could be spotted by clerical review as correctly generic), we identified new rules to filter-out non-product items (e.g. baby monitors, or shower accessories) on the unbranded category. We also dismissed by default the data on some brands (e.g. Omron, a popular blood pressure monitor producer). Taken together, the large amount of time spent on data cleaning for precise brand assignment, involving a clerical/human-in-the-loop component, represents the aspect of our solution that is most difficult to generalize across datasets.

Following brand assignment, we proceeded to *model assignment*. Unlike the case of brands, the list of possible models can be expected to be longer. For this we designed a four-step algorithm based on information propagation. The underlying idea is to propagate information which is certain, before less certain one. We present the steps in the following paragraphs. The effectiveness of its steps, for the monitor dataset, is summarized in Table 2.

(1) For a list of items from a brand, the first step is to collect likely model keywords from fields identified as good candidates to contain model information (i.e, model, model name, product model, product name, mpn, model number, mfr part number, series, or specifications). For the data extraction we use regex patterns that match on mixes of numerical/alphabetical sequences, and that are different from MPN fields or measurements. For improving the extraction we used some hard-coded rules, such as requiring the matched keyword to also appear in the page title. After this stage we can identify 2,594 possible models, covering 7,226 items from our dataset, with only 77 models reporting 10 or more products with the given model in the dataset.

(2) In the second step we sort the models identified per brand, according to their popularity, and we search products in the

Stages:	1	2	3	4
# of models	2,594	2,594	4,681	4,477
# of items with models assigned	7,226	12,103	15,112	15,722
# of models with more than 10 items	77	303	313	319

Table 2: Effectiveness of steps in our proposed information propagation-based method for model extraction

brand without a model assigned, plus those assigned from less-certain rules, to check whether the popular models are mentioned in the page title or model fields of these products. We then propagate the model assignment accordingly. Through this process our model number remains unchanged, but the number of products with assignments nearly climbs to 12,103, with 303 models having 10 or more products, and the most popular being HP's EliteDisplay E231 monitor, matching for 73 products.

(3) As a third step we extract keywords based on rules (as opposed to matching known models) from the page titles, plus letting less certain items change their assignments, according to popularity shifts. For the dataset we study, through this step we find 4,681 models, covering 15,112 products, with 313 models having 10 or more products.

(4) We conclude our proposed method by using the extracted models, sorted by popularity, for matching on non-assigned products across all valid fields (i.e., removing fields such as payment information, or return policies). At this stage we also include a voting-based extraction for potential models, some evident domain-specific cleaning of our assignments (e.g. removing common false positives, like 1080p), and attempts to properly establish when missing model assignments are still correct (i.e., items should not be assigned to a model if the information is truly absent). By the end we have 4,477 models, covering 15,722 items, with a remainder of 530 items missing a model assignment, and only 319 models having 10 or more products in the dataset.

Concerning ER, for this dataset the brands and the model assignments act as a blocking method, reducing the number of comparisons required to match items. Traditionally, statistical predictive models based on the data would be pertinent at this stage. However, due to the fact that brands and models are established in a way that is less uncertain, while uniquely determining an entity for our dataset, we found that applying ML at this stage was unnecessary, with simple matching sufficing. Thus, we consider as entities all items matching simultaneously on brand and model, reporting 90,022 matching pairs, leading to a competitive f1 score of 0.921. Altogether our solution is able to run in a short amount of time, taking less than 15 minutes on a naive configuration, not optimized for running time performance. Further tuning of rules, and the study of the bottom-line contributions (on the held-out labelled data) of the individual design choices, remain areas for future work.

Supervised-Learning Approach: For a supervised-learning perspective we deployed a contextual embedding model, BERT, to extract general representations for our product data, removing redundant text, but disregarding the schema information beyond page title (i.e., no SM). Next, we used the averaged generated embeddings per product, coupled with the ground truth on items that should be in the same block and items that perhaps should not be in the same one, as starting points for training a triplet-based learned hashing model (this is an approach stemming from multimedia data management, and showing promising early results in internal research, for more relational ER datasets like Amazon-Walmart). For the matching itself, we devised the use of a set of general classifiers, which enabled us to reason on the most promising supervised learning class of model for the matching itself. Orthogonally, we developed weak supervision rules using Snorkel[18], to filter-out products from non-product data. Unfortunately, the numerous configurable steps of this pipeline resulted non-trivial to adapt to our dataset within the short time of the challenge, and the approach did not produce entirely reasonable answers when moving beyond the training data. Thus, further work would be needed to properly adapt such pipeline to the dataset challenges. A core design choice here is to regulate the extent of domain-specific cleaning/extraction rules incorporated. As stated previously, when a sufficient amount of cleaning/extraction is done, ML can result unnecessary.

3.2 Schema Matching

Domain-Knowledge Approach: We propose a solution that aims to group the site attributes by similarity, before assigning them to a specific output class. To this end, we start by finding a token-based representation for each instance value given to a site/source attribute. We do this through English language stemming (though our dataset also includes a significant amount of text in Italian), and TF-IDF tokenization. After filtering out the most infrequent tokens (used less than 5 times) we can generate for each source attribute a histogram marking the likelihood of a given token being employed in instance-values. For our case we used 10,393 tokens. Thanks to this featurization, we could compare all pairs of source attributes in a brute-force manner, using cosine similarity. We should note that this approach for comparing distributions is generally inspired by the work of Zhang et al.[21], where authors use the Earth Mover's Distance. From our work the task of systematically determining the role of the similarity measure, all remaining things fixed, remains open. Other than matching by setting high thresholds of 0.75 (for cosine similarity), we also created filtering rules to dismiss false matches, based on syntactical similarity or hard-coded domain-specific restrictions (e.g. though vertical and horizontal frequencies might have similar histograms, they should not be matched due to their conceptually different dimensions). Brute-force matching resulted, of course, in a large computation overhead, taking up to 6 hours on non-optimized configurations.

Following the time-consuming comparisons, we require three further steps to produce results. First, some small syntax similarity-based rules to serve source attributes that remained unmatched to others. Second, a grouping procedure able to form potentially overlapping clusters of source attributes. We develop a method whereby all pairs that are connected in the shape of a complete graph (i.e. with each source attribute in the cluster graph having a connection to all the remaining) are assigned to a cluster. We also propose some rules to merge many clusters when they only differ by a few nodes, compensating for uncertainty in the rules for threshold assignment in cosine similarity matching. Finally, we need to assign each cluster to a target/mediated attribute. In absence of further information, we rely solely on the highest syntactic similarity to an item in the cluster. We could have used the ground truth to a larger extent, but dismissed this for better adaptation to the competition category. By the end, the method developed was able to map only a small set of 1,374 attributes (out of the 3,715 present in the dataset), achieving a low f1-measure of 0.316 on the held-out data. Results show that there is still a need for correcting and further improving the configuration of our proposed process; in specific, reducing the time-consuming comparisons, forming clusters and evolving the precise methods for matching clusters to target labels. Obraczka et al., in their solution for a previous DI2KG challenge describe a graph-based approach to schema matching [15] which could be adapted for study alongside our solution.

Supervised-Learning Approach: For this category we studied a semi-supervised learning method (not to be confused with active learning). We specifically adapted the K-means clustering algorithm. We started by creating a TF-IDF representation of all labelled instances, paired with their hand-crafted features (e.g. is boolean). We clustered them with K means, specifying k as the number of expected labels. After computing the centroids we were able to assign, by their similarity, all unlabeled items to clusters. This enabled us to determine the top words per cluster, helping us to featurize our labeled data anew. What we devised as a reasonable step to follow is an iterative process where the vocabularies of common words in a cluster are updated, making us change representations for the labeled data, and the clustering is performed again, until convergence. Through this approach we reached an f1 score of 0.804 on the training data, but difficulties in generalizing to the competition dataset. Similar observations were found for the notebook dataset.

4 CONCLUSION & TAKEAWAYS

In this paper we evaluate proposed solutions for ER and SM, on a challenge dataset, with the proposals for ML-ER and SM being less successful and thus, indicating a need for follow-up improvements.

Regarding the ML approaches, our current observations (omitting from this work comprehensive model tuning and introspection) remain preliminary and inconclusive. Within this scope of understanding, we found that applying out-of-the-box ML models (e.g. Bert embeddings, cosine similarity comparisons and general classifiers) to a dataset with a highly heterogeneous schema and noisy data (e.g. featuring redundant publicity text) was not immediately appropriate for generalizing beyond the labelled data. This is an interesting observation, given that similar integration applications report more success in datasets with noisy data but more homogeneous schemas [5, 14]. Our observation concurs with the theorem, whereby we find that there is no free lunch: beyond simply choosing competitive model classes, a precise amount of problem framing/hypothesis-space adaptation are very much required for successful learning. In the direction of such adaptation efforts in data cleaning/extraction (shown in Fig. 1) can be expected to ease integration tasks (we report a case for ER, where we find that sufficient cleaning/extraction makes the ER problem trivial to solve), but such efforts can be administered in the form of fixes and extensive domain tuning that reduce the generality of the solution, making it hard for efforts done for one dataset to work for another. All things equal, holistic data integration would benefit the most from tools that reduce the efforts and facilitate the integration tasks, without compromises in their generalization power across datasets and application scenarios.

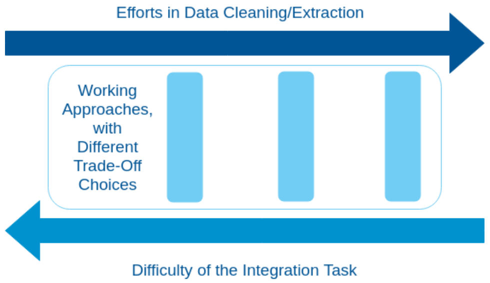
Efforts in Data Cleaning/Extraction
Working Approaches, with Different Trade-Off Choices
Difficulty of the Integration Task

Figure 1: Efforts in data cleaning/extraction can be expected to reduce the difficulty of data integration. All things equal, for approaches that solve a task, preferred solutions should reduce efforts without losing generality.

Regarding our domain-specific solutions, we conceive that future work comparing their results with those of state-of-the-art tools (e.g. [4, 15, 17]), could improve our understanding of limitations and advantages.

Although both domain-specific approaches described in this paper are different, they share two common features: on the one hand, their reliance on *domain-specific tuning* (e.g. rules for brand extraction considering alternative names, model extraction tweaking for fields where the model might be mentioned, or schema matching rules to enforce should-not-link constraints on similar-yet-different attributes, like horizontal and vertical frequency); on the other hand, their use of heuristics involving *information propagation*. In the case of ER, we employ the latter explicitly to assign values extracted from certain rules to less certain ones, enabling the process to be guided by consensus on the most popular extracted values. For SM, information propagation is relevant to decide whether complete graphs of items that have a high similarity in a pair-wise fashion should be combined. We consider that our experience serves as a report confirming the utility of these 2 kinds of solution features for integration tasks.

Based on our experience with the domain-specific solutions we venture two suggestions for similar integration work dealing with schema heterogeneity and noisy data. First, that methods to standardize and exploit better domain knowledge, bringing also the human into the loop, are truly needed (i.e., an aspect that could help to generalize across datasets the data cleaning/extraction rules, or tackle the zero-shot learning problem at the heart of SM ground truths lacking examples for some target attributes). Second, that to capture and improve the useful algorithmic choices based on information propagation that we employed, extending collective & graph-based methods [19] (e.g. framing ER as a link prediction problem to evaluate with graph neural networks), combined with the state-of-the-art in attribute/similarity representation learning, could be a good way forward.

5 ACKNOWLEDGMENTS

The authors would like to express their gratitude to the organizers of the Second DI2KG workshop and challenge. The authors would like also to thank Vishnu Unnikrishnan and Xiao Chen, for nurturing discussions on data integration. Finally, the authors would like to thank Bala Gurumurthy, plus participants and organizers of the DBSE Scientific Team Project SS2020, at the University of Magdeburg, for useful comments during the progress of this work.

REFERENCES

[1] Philip A Bernstein, Jayant Madhavan, and Erhard Rahm. 2011. Generic schema matching, ten years later. *Proceedings of the VLDB Endowment* 4, 11 (2011), 695–701.

[2] Ursin Brunner and Kurt Stockinger. 2020. Entity Matching with Transformer Architectures - A Step Forward in Data Integration. In *Proceedings of the 23nd International Conference on Extending Database Technology, EDBT 2020, Copenhagen, Denmark, March 30 - April 02, 2020*, Angela Bonifati, Yongluan Zhou, Marcos Antonio Vaz Salles, Alexander Böhm, Dan Olteanu, George H. L. Fletcher, Arijit Khan, and Bin Yang (Eds.). OpenProceedings.org, 463–473. https://doi.org/10.5441/002/edbt.2020.58

[3] Hong-Hai Do and Erhard Rahm. 2002. COMA—a system for flexible combination of schema matching approaches. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*. Elsevier, 610–621.

[4] AnHai Doan, Pradap Konda, Paul Suganthan GC, Yash Govind, Derek Paulsen, Kaushik Chandrasekhar, Philip Martinkus, and Matthew Christie. 2020. Magellan: toward building ecosystems of entity matching solutions. *Commun. ACM* 63, 8 (2020), 83–91.

[5] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. 2017. DeepER–Deep Entity Resolution. *arXiv preprint arXiv:1710.00597* (2017).

[6] IP Fellegi and AB Sunter. 1969. A theory of record linkage, American Statistical Association Journal, vol. 64.

[7] Raul Castro Fernandez, Essam Mansour, Abdulhakim A Qahtan, Ahmed Elmagarmid, Ihab Ilyas, Samuel Madden, Mourad Ouzzani, Michael Stonebraker, and Nan Tang. 2018. Seeping semantics: Linking datasets using word embeddings for data discovery. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 989–1000.

[8] Vijay Gadepally, Justin Goodwin, Jeremy Kepner, Albert Reuther, Hayley Reynolds, Siddharth Samsi, Jonathan Su, and David Martinez. 2019. AI Enabling Technologies: A Survey. *arXiv preprint arXiv:1905.03592* (2019).

[9] Enrico Gallinucci, Matteo Golfarelli, and Stefano Rizzi. 2018. Schema profiling of document-oriented databases. *Information Systems* 75 (2018), 13–25.

[10] Alon Halevy. 2009. *Information Integration*. Springer US, Boston, MA, 1490–1496. https://doi.org/10.1007/978-0-387-39940-9_1069

[11] Jingya Hui, Lingli Li, and Zhaogong Zhang. 2018. Integration of big data: a survey. In *International Conference of Pioneering Computer Scientists, Engineers and Educators*. Springer, 101–121.

[12] Evgeny Krivosheev, Mattia Atzeni, Katsiaryna Mirylenka, Paolo Scotton, and Fabio Casati. 2020. Siamese Graph Neural Networks for Data Integration. *arXiv preprint arXiv:2001.06543* (2020).

[13] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. 2002. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Proceedings 18th International Conference on Data Engineering*. IEEE, 117–128.

[14] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data*. 19–34.

[15] Daniel Obraczka, Alieh Saeedi, and Erhard Rahm. 2019. Knowledge graph completion with FAMER. *Proceedings of the DI2KG* (2019).

[16] George Papadakis, Ekaterini Ioannou, and Themis Palpanas. 2020. Entity Resolution: Past, Present and Yet-to-Come: From Structured to Heterogeneous, to Crowd-sourced, to Deep Learned. In *EDBT/ICDT 2020 Joint Conference*.

[17] George Papadakis, George Mandilaras, Luca Gagliardelli, Giovanni Simonini, Emmanouil Thanos, George Giannakopoulos, Sonia Bergamaschi, Themis Palpanas, and Manolis Koubarakis. 2020. Three-dimensional Entity Resolution with JedAI. *Information Systems* (2020), 101565.

[18] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, Vol. 11. NIH Public Access, 269.

[19] Alieh Saeedi, Eric Peukert, and Erhard Rahm. 2020. Incremental Multi-source Entity Resolution for Knowledge Graph Completion. In *European Semantic Web Conference*. Springer, 393–408.

[20] Saravanan Thirumuruganathan, Nan Tang, Mourad Ouzzani, and AnHai Doan. 2020. Data Curation with Deep Learning.. In *EDBT*. 277–286.

[21] Meihui Zhang, Marios Hadjieleftheriou, Beng Chin Ooi, Cecilia M Procopiuc, and Divesh Srivastava. 2011. Automatic discovery of attributes in relational databases. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. 109–120.