

Neural Data Search for Table Augmentation

Alexander Brinkmann^{1,2}

¹Data and Web Science Group, University of Mannheim, B6, 26, 68159 Mannheim, Germany

²supervised by Christian Bizer

Abstract

Tabular data is widely available on the web and in private data lakes run by commercial companies or research institutes. However, data that is essential for a specific task at hand is often scattered throughout numerous tables in these data lakes. Accessing this data requires retrieving the relevant information for the task. One approach to retrieve this data is through table augmentation. Table augmentation adds an additional attribute to a query table and populates the values of that attribute with data from the data lake. My research focuses on evaluating methods for augmenting a table with an additional attribute.

Table augmentation presents a variety of challenges due to the heterogeneity of data sources and the multitude of possible combinations of methods. To successfully augment a query table based on tabular data from a data lake, several tasks such as data normalization, data search, schema matching, information extraction and data fusion must be performed. In my work, I empirically compare methods for data search, information extraction and data fusion as well as complete table augmentation pipelines using different datasets containing tabular data found in real-world data lakes. Methodologically, I plan to introduce new neural techniques for data search, information extraction and data fusion in the context of table augmentation. These new methods, as well as existing symbolic data search methods for table augmentation, will be empirically evaluated on two sets of benchmark query tables. The aim is to identify task- and dataset-specific challenges for data search, information extraction and data fusion methods. By profiling the datasets and analysing the errors made by the evaluated methods on the test query tables, the strengths and weaknesses of the methods can be systematically identified. Data search and information extraction methods should maximize recall while data fusion methods should achieve high accuracy. Pipelines built on the basis of the new methods should deliver their results quickly without compromising the highest possible accuracy of the augmented attribute values.

Keywords

Table Augmentation, Data Search, Information Extraction, Data Fusion

1. Introduction

Tabular data is widely available on the web and in private data lakes run by research institutes or companies. Data that is relevant to a particular task is scattered across multiple tables in these data lakes. Using a data lake requires retrieving the relevant data for the task at hand. There are several approaches to searching tabular data in these data lakes, such as Google's Dataset Search, which relies on a keyword search that exploits a table's metadata to find relevant tables [1]. Recognising the heterogeneity and scarcity of metadata, table augmentation pipelines explore data-driven search beyond keyword search [2, 3]. For table augmentation, a user provides an initial query table. This query table can be augmented by adding new columns, new rows, and completing cells with relevant data from a data lake [4]. In my research, I focus on augmenting a query table with a new column that is populated with content from a table corpus also known as augmentation by attribute name [2]. The query table and column header are user-defined. Figure 1 shows an example. Cell completion is closely related but is not the

main focus of my work. I do not cover the addition of rows to a query table. For the remainder of this proposal, table augmentation by attribute name [2] will be referred to as table augmentation.


Depending on the query table and the heterogeneous content of the table corpus, table augmentation pipelines need to address challenges like data normalization, data search, schema matching, information extraction, and data fusion to augment a query table with a new attribute. Methodologically, I will present new neural methods for data search, information extraction and data fusion in the context of table augmentation. The methods will be compared with existing methods from related work. When it is necessary to complete a pipeline I rely on existing methods. Table augmentation pipelines should augment the query table with the correct values, resulting in high accuracy. If two pipelines augment a query table with the same accuracy, the pipeline that delivers the result faster is preferred. New methods for the intermediate tasks of data search and information extraction are benchmarked against the runtime reduction of complete pipelines, which should not negatively affect the accuracy of the pipeline.

All methods and complete pipelines are evaluated on the schema.org table corpus¹, and a Web Tables corpus [5, 6]. The datasets represent the heterogeneity

Published in the Workshop Proceedings of the EDBT/ICDT 2023 Joint Conference (March 28-March 31, 2023, Ioannina, Greece)

✉ alexander.brinkmann@uni-mannheim.de (A. Brinkmann)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

¹<http://webdatacommons.org/structureddata/schemaorgtables/>

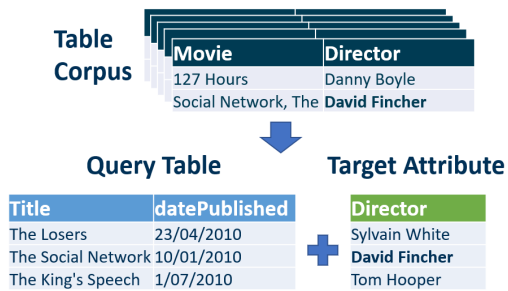


Figure 1: Table Augmentation: Given a table corpus, a query table and a target attribute, fill the attribute values of the target attribute from the table corpus.

of data sources for tabular data as it occurs in real-world data lakes. For the empirical evaluation, I will define a set of benchmark query tables including ground truth information for each of the datasets if the necessary resources are not provided in the related work [3]. Through the benchmark query tables, all methods are compared on a common basis. By profiling the datasets and systematically analysing the errors made by the evaluated methods, the strengths and weaknesses of the methods in terms of pipeline accuracy and pipeline runtime can be identified.

With my PhD research, I will make the following contributions:

- Introduction of neural methods for data search, information extraction and data fusion in the context of table augmentation.
- Introduction of benchmark query tables for the table corpora schema.org table corpus and Web Tables.
- Evaluation of new methods, existing methods and complete pipelines for table augmentation on benchmark query tables in terms of accuracy and runtime to systematically identify strengths and weaknesses.

In this paper, I outline my plan to achieve these contributions. The paper is organised as follows. Section 2 discusses related work. My general work plan is given in Section 3. Section 4 introduces work that has already been done and Section 5 concludes my research plans.

2. Related Work

Table augmentation is a long-standing challenge in industry and academia that has been addressed by several previous works [2, 3, 7]. Octopus [7], Infogather [2] and the Mannheim Search Join Engine [3] implement table augmentation pipelines that rely on symbolic methods

for tasks such as data search and entity matching. Entity matching aims to identify records in two datasets that describe the same real-world entity [8]. To reduce these runtimes, entity resolution pipelines consist of two parts: a blocker, which applies a computationally cheap method to select candidate pairs of records, and a matcher, which then extracts matching pairs from this set using more expensive methods [8]. Both blocking [9, 10] and matching [11, 12, 13] have recently been successfully tackled with deep learning approaches. My research focuses on experimenting with neural and symbolic methods that find a matching record for each record in the query table. Another approach to finding matching records in the data lake is to search for joinable tables based on an explicitly mentioned column [14]. If a joinable table contains the target attribute value being searched for, the query table is populated accordingly.

3. Work Plan

This section presents my work plan. To provide a common ground for evaluating table augmentation methods, I present table corpora that differ in size, source, and content representing real-world data lakes. I then discuss the table augmentation methods for Data Search, Information Extraction and Data Fusion I will experiment with.

3.1. Table Corpora for Evaluation

I evaluate table augmentation methods on the large-scale table corpora WDC Table Corpus and Web Tables. Through the diversity of the table corpora, I aim to represent the heterogeneity of tabular data present in real-world data lakes. For this evaluation, the table corpora are profiled to identify dataset-specific challenges and a set of benchmark query tables will be defined in order to compare the table augmentation methods on a common basis.

Web Data Commons Schema.org Table Corpus The WDC table corpus² consists of 4.2 million relational tables generated by extracting schema.org³ annotations from the Common Crawl and grouping the annotations by class and host. All tables in this corpus share a common schema. By removing schema matching a focus can be put on other tasks of the table augmentation pipeline.

WDC WebTables and Dresden Web Tables WDC WebTables and Dresden Web Tables contain 59M to 90M relational tables extracted HTML tables in the Common Crawl [5, 6]. The heterogeneity of tables and their usage in related work make the table corpora interesting for my research [4].

²<http://webdatacommons.org/structureddata/schemaorgtables/>

³<https://schema.org/>

3.2. Table Augmentation pipeline

Depending on the content of the query table and the table corpus, table augmentation pipelines have to deal with data search, schema matching, information extraction and data fusion [4]. Table 1 shows my timeline to work on the specific tasks.

Table 1
PhD work plan.

Task	Start	End
Neural Entity Search for Table Augmentation	Apr '22	Feb '23
Information Extraction on the WDC Corpus	Mar '23	Jun '23
Data Fusion of Records from the WDC Corpus	Jul '23	Oct '23
Information Extraction on the Web Table Corpus	Nov '23	Feb '24
Data Fusion of Records from the Web Table Corpus	Mar '24	Jun '24

Data Search. Data search and more specifically record search aims to find records in a data lake, which match the records of a query table. Two records match if they describe the same real-world entity. Record search methods exploit schema, context and content of a query table to find matching records in a table corpus [4, 15]. Existing symbolic methods use approaches like calculating the edit-distance of table headers and the Jaccard similarity of two records [3] or measuring the similarity through the vector product of TF/IDF-weighted term vectors to find matching records [2]. Existing neural methods embed table records into a high-dimensional space and apply nearest neighbour search to retrieve matching records for query table records [9, 10]. My initial experiments as presented in Section 4 deal with entity matching pipelines.

Information Extraction. Information extraction is only relevant if the content of the target attribute is not explicitly shared in the table corpus, but has to be extracted from another attribute. For example, if a user searches for the colours of a set of products, this information may be contained in the attribute description and needs to be extracted before the target attribute colour can be populated in the query table. I will experiment with approaches that finetune large language models (LLM) to extract attribute-value pairs [16] and in-context learning where a LLM predicts the attribute-values pairs based on a context augmented with a few examples [17]. If the extracted attributes do not match the target attribute, schema matching might be helpful to match extracted attributes to the target attribute. In this context, I will test label-based, instance-based, structure-based, or combined methods for schema matching [18, 19, 20].

Data Fusion. Data search delivers lists of matching records for the query table records. Augmenting a record based on matching records can lead to data conflicts. Data fusion tries to solve these conflicts. Classic conflict resolution methods are instance- or metadata-based. Examples of instance-based methods are majority vote or averaging conflicting values [8]. Source quality [21] and minimal set coverage [22] are metadata-based and exploit provenance information to resolve conflicts. I plan to compare the classic methods to generative LLMs [23] and retrieval-augmented generative LLMs [24]. Generative LLMs memorize knowledge from a data lake during training and directly predict values of the target attribute [23]. For the retrieval-augmented generative LLMs, the retrieved conflicting records are added to the context of the generative LLMs, which resolve the conflict by predicting the values of the target attribute [24]. Both approaches have been successfully applied to related NLP tasks [23, 24] and are promising for data fusion, too.

4. Initial Experiments

My initial experiments deal with entity resolution pipelines that aim to identify records across two datasets that describe the same real-world entity [8]. Since comparing all record pairs between two datasets can be computationally expensive, entity resolution is usually tackled by a blocking and a matching step. Blocking applies a computationally cheap method to remove non-matching record pairs and produce a smaller set of candidate record pairs reducing the workload of the matcher. During matching a more expensive pair-wise matcher generates a final set of matching record pairs. In the context of these experiments, I propose SC-Block, a supervised contrastive blocking method which combines supervised contrastive learning for positioning records in an embedding space and nearest neighbour search for candidate set building. In addition to pairs completeness and candidate set size, I report F1 scores and runtimes of complete entity resolution pipelines. I do this to evaluate SC-Block's impact on complete entity resolution pipelines. SC-Block is benchmarked against eight state-of-the-art blockers and combined with four state-of-the-art matchers. On three product-matching datasets from related work [9, 10], SC-Block creates the smallest candidate sets and pipelines with SC-Block run 1.5 to 2 times faster compared to the benchmarked blockers without affecting the F1 score of the pipeline. These datasets are rather small, which might lead to runtime effects resulting from a large vocabulary size being overlooked. In order to measure runtimes in a more challenging setting, I introduce a new benchmark dataset featuring a large vocabulary of terms used within entity descriptions. On this large-scale benchmark dataset, pipelines utilizing SC-Block and the best-

performing matcher execute 8 times faster than pipelines utilizing the second best-performing BM25 blocker with the same matcher reducing the runtime from 2.5 hours to 18 minutes, clearly compensating for the 5 minutes that are required for training SC-Block. These results are promising for table augmentation because they show how the symbolic data search in related work [2, 3] can be accelerated on large-scale datasets.

5. Conclusion

In this proposal, I have outlined the research for my PhD. I will present new neural methods within table augmentation pipelines. The methods will be benchmarked against existing methods on two sets of benchmark query tables defined to compare complete table augmentation pipelines and methods for specific table augmentation tasks on a common basis. The goal of complete pipelines is to augment a query table with high accuracy. In addition, I will compare the runtimes of complete pipelines and evaluate how new methods for data search and information extraction can reduce the overall runtime without negatively affecting the accuracy of the pipeline. By profiling the benchmark query tables and their corresponding table corpora, I will identify the strengths and weaknesses of both complete table augmentation pipelines and methods for specific pipeline tasks.

References

- [1] N. Noy, M. Burgess, D. Brickley, Google Dataset Search: Building a search engine for datasets in an open Web ecosystem, in: WebConf '19, 2019.
- [2] M. Yakout, K. Ganjam, K. Chakrabarti, InfoGather: entity augmentation and attribute discovery by holistic matching with web tables, SIGMOD '12, 2012, pp. 97–108.
- [3] O. Lehmborg, D. Ritze, P. Ristoski, The Mannheim Search Join Engine, Journal of Web Semantics 35 (2015) 159–166.
- [4] S. Zhang, K. Balog, Web Table Extraction, Retrieval, and Augmentation: A Survey, ACM Trans. Intell. Syst. Technol. 11 (2020) 1–35.
- [5] O. Lehmborg, D. Ritze, R. Meusel, A Large Public Corpus of Web Tables containing Time and Context Metadata, in: WWW '16, 2016, pp. 75–76.
- [6] J. Eberius, K. Braunschweig, M. Hentsch, Building the Dresden Web Table Corpus: A Classification Approach, in: BDC '15, 2015, pp. 41–50.
- [7] M. J. Cafarella, A. Halevy, N. Khoussainova, Data integration for the relational web, VLDB '09 2 (2009) 1090–1101.
- [8] P. Christen, Data matching : concepts and techniques for record linkage, entity resolution, and duplicate detection, Springer, Berlin, Heidelberg, 2012.
- [9] S. Thirumuruganathan, H. Li, N. Tang, Deep learning for blocking in entity matching: a design space exploration, VLDB 2021 (2021) 2459–2472.
- [10] R. Wang, Y. Li, J. Wang, Sudowoodo: Contrastive Self-supervised Learning for Multi-purpose Data Integration and Preparation, 2022. ArXiv:2207.04122.
- [11] Y. Li, J. Li, Y. Suhara, A. Doan, W.-C. Tan, Deep Entity Matching with Pre-Trained Language Models, VLDB 2020 14 (2020) 50–60. ArXiv: 2004.00584.
- [12] U. Brunner, K. Stockinger, Entity matching with transformer architectures - a step forward in data integration, in: EDBT 2020, 2020.
- [13] R. Peeters, C. Bizer, Dual-objective fine-tuning of BERT for entity matching, in: VLDB 2021, volume 14 10, New York, NY, 2021, pp. 1913–1921.
- [14] E. Zhu, D. Deng, F. Nargesian, R. J. Miller, JOSIE: Overlap Set Similarity Search for Finding Joinable Tables in Data Lakes, SIGMOD '19, New York, NY, USA, 2019, pp. 847–864.
- [15] N. Barlaug, J. A. Gulla, Neural Networks for Entity Matching: A Survey, ACM TKDD 15 (2021) 52:1–52:37.
- [16] X. Zhang, C. Zhang, X. Li, OA-Mine: Open-World Attribute Mining for E-Commerce Products with Weak Supervision, in: WWW '22, ACM, Virtual Event, Lyon France, 2022, pp. 3153–3161.
- [17] Q. Dong, L. Li, D. Dai, A Survey on In-context Learning, 2023. ArXiv:2301.00234 [cs].
- [18] E. Rahm, P. A. Bernstein, A survey of approaches to automatic schema matching, VLDB '01 (2001) 334–350.
- [19] C. Koutras, G. Siachamis, A. Ionescu, Valentine: Evaluating Matching Techniques for Dataset Discovery, in: ICDE '21, 2021, pp. 468–479.
- [20] R. Shraga, A. Gal, H. Roitman, ADnEV: cross-domain schema matching using deep similarity matrix adjustment and evaluation, VLDB '20 (2020) 1401–1415.
- [21] X. L. Dong, E. Gabrilovich, K. Murphy, Knowledge-based trust: estimating the trustworthiness of web sources, VLDB '15 (2015) 938–949.
- [22] J. Eberius, M. Thiele, K. Braunschweig, Top-k entity augmentation using consistent set covering, in: SSDBM '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 1–12.
- [23] C. Raffel, N. Shazeer, A. Roberts, Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, Machine Learning Research 21 (2020) 1–67.
- [24] P. Lewis, E. Perez, A. Piktus, Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, in: NeurIPS 2020, volume 33, 2020, pp. 9459–9474.